

# Trabajo Práctico 1. Programación Orientada a Objetos

## Puntuación

Puntaje Total: 100 puntos

Aprobación: 60 puntos

Fecha de entrega: 05/10/2025 – 23:59 Hs.

## Condiciones de entrega

1. El presente trabajo práctico deberá resolverse en grupos de cuatro (4) alumnos.
2. Entrega: Se realizará por medio del Campus Virtual de la UTN, en la tarea correspondiente al TP 1. La extensión del archivo será .zip o .rar, de acuerdo al programa de compresión usado. El nombre del archivo se consigue concatenando un prefijo del número del TP con los apellidos de los integrantes por orden alfabético y separados por guiones (por ejemplo: para apellidos Pérez, Acevedo y González, el nombre será: TP1-Acevedo-Gonzalez-Perez.zip). Note que no hay espacios en blanco ni acentos en el nombre de archivo. Dentro del archivo de entrega, deben constar los siguientes:
  - **Fuentes Pharo Smalltalk:** Se debe crear un paquete para todas las clases del TP, llamado TP1. Realizar un #fileOut de este paquete y guardarlo en un archivo TP1.st
  - Los **Casos de Prueba** se entregarán en un archivo de texto, no deben ser capturas de pantalla. Deberán cubrir diferentes resultados que puedan obtenerse según las funcionalidades solicitadas. Se enfatiza que se adjunten casos de prueba que sean claros, válidos y suficientes para poder probar el trabajo. Como sugerencia puede ir acumulando los pruebas en un Playground y luego grabarlo a disco con el nombre casos-de-prueba.txt.
  - **Utilización de modelos de lenguaje (LLMs):** en caso de utilizar estas herramientas se entregará un archivo de texto (.txt) o markdown (.md o .markdown) con un listado describiendo brevemente los componentes o funcionalidades que se analizaron o resolvieron mediante conversaciones con modelos de IA generativa (ChatGTP, Gemini, etc.). Incluir al final de este archivo una sección con conclusiones. Archivo: consultas-IA.markdown.
  - **Archivo de integrantes** (integrantes.txt) con una línea para cada integrante en la cual figure el nombre del alumno/a y su dirección de email.
3. Penalización por entrega fuera de término: Si el trabajo práctico se entrega después de la fecha indicada, y hasta una semana tarde, tendrá una quita de 15 puntos. No serán recibidos trabajos luego de una semana de la fecha de entrega. Los trabajos prácticos para los cuales se solicite rehacer/corregir fuera de la fecha de entrega tienen una quita de 30 puntos.

## Introducción

La insuficiencia renal crónica es una enfermedad que deteriora de forma progresiva e irreversible el funcionamiento de los riñones. Para muchos **pacientes**, el trasplante renal constituye la mejor alternativa terapéutica, ya que les permite recuperar su calidad de vida y, en muchos casos, salvarla.

El principal desafío de un trasplante es contar con un donante compatible. Si el órgano no es adecuado para el receptor, el sistema inmunológico lo rechaza y el resultado puede ser muy grave. En algunos casos, un paciente tiene un familiar o allegado dispuesto a donar. Si ambos son compatibles, la cirugía puede realizarse sin mayores dificultades. Sin embargo, no siempre se da esa situación: puede ocurrir que el **donante** no sea compatible con su pariente, pero sí lo sea con otro paciente en lista de espera.

De aquí surge la importancia de **evaluar de manera sistemática la compatibilidad entre donantes y receptores**. Para ello se utilizan distintos estudios clínicos y criterios médicos. Algunos criterios son obligatorios: si no se cumplen, el trasplante no puede realizarse. Otros criterios son deseables: permiten comparar a varios donantes posibles y decidir cuál ofrece la mejor opción para un receptor dado.

En este trabajo práctico, el objetivo es **modelar e implementar en Pharo (Smalltalk) un sistema de gestión para un centro de trasplantes, que represente pacientes, donantes y estudios clínicos, y que permita determinar automáticamente la compatibilidad entre ellos**.

A fin de simplificar el trabajo, muchos detalles referidos al problema son obviados en este enunciado. Es importante remarcar que, antes de cualquier aplicación práctica de la herramienta, será necesario un análisis más detallado y validado por expertos de los criterios médicos aquí presentados.

## Caracterización de criterios de compatibilidad y estudios clínicos necesarios

Para determinar si un trasplante puede realizarse de manera segura y eficiente, se deben evaluar distintos **criterios de compatibilidad** entre **donante y receptor**. Algunos de estos criterios son **obligatorios**, lo cual significa que si no se cumplen el trasplante no puede realizarse; otros son **deseables**, y sirven para priorizar qué donante es más adecuado cuando existen varias opciones. Cada criterio se evalúa a partir de la información del paciente, del donante, o de ambos, pudiendo incluir uno o más **estudios clínicos**, cuyos resultados proporcionan la base necesaria para calcular la compatibilidad.

La siguiente tabla resume los criterios, su clasificación, descripción y estudios clínicos asociados:

Criterio	Clasificación	Descripción breve	Estudios clínicos necesarios
Grupo sanguíneo (ABO)	Obligatorio	Compatibilidad de grupos sanguíneos entre donante y receptor.	Análisis de sangre donante y paciente
Crossmatch	Obligatorio	Determina si existen anticuerpos del receptor que atacan las células del donante; un resultado positivo indica alto riesgo de rechazo.	Prueba de crossmatch linfocitotóxico o por citometría de flujo
Serologías activas	Obligatorio	Detecta infecciones activas que contraindican el trasplante.	HBsAg, HCV RNA/Ag, HIV (donante y receptor)
Estado clínico del donante	Obligatorio	Evalúa la aptitud general del donante para someterse a la cirugía y donar un riñón.	Historia clínica, función renal residual, presión arterial, comorbilidades relevantes
HLA	Deseable	Mide la compatibilidad del sistema HLA; menor número de "mismatches" mejora la probabilidad de éxito.	Tipificación HLA A, B y DR de donante y paciente
cPRA / PRA	Deseable	Indica el nivel de sensibilización del receptor; menor cPRA implica mayor probabilidad de éxito.	HLA del donante, cPRA o PRA del paciente
Tamaño corporal	Deseable	Compatibilidad entre el tamaño del riñón donado y el receptor; diferencia menor mejora la función post-trasplante.	Altura y peso de donante y receptor (para estimar IMC o masa renal)
Edad	Deseable	Diferencia de edad entre donante y receptor; menor diferencia puede ser preferible.	Edad del donante y del paciente

FIN

La siguiente tabla resume las reglas posibles a aplicar para cada criterio:

Criterio	Evaluación de compatibilidad
ABO (Obligatorio)	<p>Se requieren análisis de sangre del paciente y del donante. Estudios: AnalisisDeSangre, con atributo grupoSanguineo.</p> <p>Regla: Los grupos sanguíneos deben ser compatibles, es decir:</p> <ul style="list-style-type: none"> <li>• O → cualquiera,</li> <li>• A → A / AB,</li> <li>• B → B / AB,</li> <li>• AB → AB</li> </ul> <p>Si son compatibles #apto, caso contrario #noApto.</p>
Crossmatch (Obligatorio)	<p>Se requiere una prueba cruzada (crossmatch) específica entre el donante y el paciente.</p> <p>Estudio: EstudioCrossmatch, con resultado #positivo o #negativo.</p> <p>Regla:</p> <ul style="list-style-type: none"> <li>• #positivo → #noApto,</li> <li>• #negativo → #apto.</li> </ul>
Serologías activas (Obligatorio)	<p>Se requiere evaluar posibles infecciones activas en el donante y el receptor que pueden contraindicar el trasplante (HBsAg, HCV, HIV, etc.)</p> <p>Estudios: EstudioSerologicoCompleto, con resultado #apto o #noApto.</p> <p>Regla: <u>ambos estudios (donante y paciente) deben indicar #apto.</u></p>
Estado clínico del donante (Obligatorio)	<p>Se requiere una evaluación general de la aptitud del donante para someterse a la cirugía y tener un riñón menos luego de la misma.</p> <p>Estudio: EstudioAptitudGeneral, con resultado #apto o #noApto.</p> <p>Regla: el resultado del estudio debe ser #apto.</p>
HLA (Deseable)	<p>Se requiere la tipificación HLA (Human Leukocyte Antigen) A, B y DR de donante y paciente.</p> <p>Estudios: EstudioTipificacionHLA, con lista de dos alelos para cada locus (A, B, DR).</p> <p><u>Ejemplo:</u> Paciente: A = {A1, A2}, B = {B7, B44}, DR = {DR1, DR4} Donante: A = {A2, A3}, B = {B7, B8}, DR = {DR4, DR15}</p>

	<p>Regla (extremadamente simplificada): Contar cuántos alelos del donante están presentes en el paciente, y dividir por el máximo de coincidencias posibles.</p> <p><u>En el ejemplo:</u></p> <ul style="list-style-type: none"> <li>• Locus A → coincidencia en A2 → 1 match</li> <li>• Locus B → coincidencia en B7 → 1 match</li> <li>• Locus DR → coincidencia en DR4 → 1 match</li> </ul> <p>Por lo tanto nro. de <u>matches</u> = 3.</p> <p>Se comparan 2 alelos por locus → <u>máximo</u> 6 matches.</p> <p>porcentaje = (nro. de matches / máximo) * 100  <math>= (3 / 6) * 100 = 50</math></p> <p>Médicamente la compatibilidad HLA es un factor determinante para predecir la supervivencia del injerto, sobre todo a largo plazo.</p> <p>Ponderación propuesta: 45%</p>
cPRA / PRA (Deseable)	<p>Se requiere evaluar el nivel de anticuerpos anti-HLA del paciente contra la tipificación HLA del donante.</p> <p>Estudios: EstudioTipificacionHLA del donante (el mismo estudio del criterio anterior) y EstudioAnticuerposPRA del paciente.</p> <p>EstudioAnticuerposPRA cuenta con un porcentaje de reactividad (presencia de anticuerpos) para distintos alelos.</p> <p><u>Ejemplo:</u> {A3 → 15, A4 → 80, B8 → 10} </p> <p>Regla (extremadamente simplificada): sumar todos los porcentajes de reactividad para los cuales el HLA del donante incluye al alelo en alguno de los loci.</p> <p>porcentaje = <math>\max(0, 100 - \text{suma})</math></p> <p><u>En el ejemplo:</u> Donante tiene alelos {A2, A3, B7, B8, DR4, DR15}</p> <ul style="list-style-type: none"> <li>• Alelo A3 → +15</li> <li>• Alelo A4 → ausente en HLA donante, no se considera.</li> <li>• Alelo B8 → +10</li> </ul> <p>Por lo tanto suma = 25.</p> <p>porcentaje = <math>\max(0, 100 - 25) = 75</math></p> <p>Médicamente este criterio es un factor determinante para predecir el riesgo de rechazo del riñón trasplantado.</p> <p>Ponderación propuesta: 45%</p>
Tamaño corporal (Deseable)	<p>Se comparan los IMC (Índices de Masa Corporal) de donante y paciente. El cálculo del IMC requiere conocer el peso y la altura:</p> <p style="text-align: center;"><u>IMC = peso [kg] / (estatura [m])<sup>2</sup></u></p>

	<p>Regla (simplificada, utilizando el valor absoluto de la diferencia entre los IMC):</p> $\text{diferencia} =  \text{IMC\_donante} - \text{IMC\_paciente} $ $\text{porcentaje} = \max(0, 1 - \text{diferencia} / 10) * 100$ <p>Ponderación propuesta: 5%</p>
Edad (Deseable)	<p>Se comparan las edades del donante y receptor.</p> <p>Regla: penalizar mayor diferencia de edad, si la diferencia es 30 años o más el porcentaje es 0.</p> $\text{diferencia} =  \text{edad\_donante} - \text{edad\_paciente} $ $\text{porcentaje} = \max(0, 1 - \text{diferencia} / 30) * 100$ <p>Ponderación propuesta: 5%</p>

## Evaluación de compatibilidades donante-paciente

Como se mencionó previamente, la evaluación de compatibilidades tendrá en cuenta distintos criterios, los cuales requieren contar con cierta información para ser evaluados. Esta información necesaria incluye datos tales como la edad, mediciones (altura, peso) y estudios clínicos que están asociados al donante y/o al paciente.

Por ejemplo, el criterio de compatibilidad de grupo sanguíneo (ABO) requiere que tanto el donante como el paciente cuenten con un AnalisisDeSangre, que brinda la información del grupoSanguineo de cada uno, y permite hacer la comparación. Aquí, AnalisisDeSangre será una subclase de EstudioClinico. La evaluación de cada criterio podrá realizarse si los estudios o información necesaria está disponible.

Si se cuenta con la información necesaria, al evaluar un criterio obligatorio se obtendrá como resultado un símbolo: #apto o #noApto. Por su parte, al evaluar un criterio deseable se obtendrá como resultado un porcentaje, representado como un número en el intervalo [0, 100]. A su vez, cada criterio deseable tendrá asociada una ponderación.

El algoritmo a utilizar puede describirse del siguiente modo:

### Algoritmo de evaluación de compatibilidad

#### 1. Entrada de datos:

- Se reciben los objetos Paciente y Donante, que tienen asociados los estudios clínicos que realizaron.
- Se tienen definidos un conjunto de criterios obligatorios (que deben cumplirse estrictamente) y un conjunto de criterios deseables (que aportan puntaje según el grado de cumplimiento).

- Cada criterio deseable tiene asignada una ponderación (peso) que representa su importancia relativa. Idealmente, la suma de las ponderaciones es 100, pero esto no es requisito para que el algoritmo funcione correctamente.

## 2. Inicialización

- Se declaran variables acumulador, sumaPonderaciones y estudiosFaltantes.
- estudiosFaltantes = una lista vacía.

## 3. Control de información y estudios clínicos faltantes

- Para cada criterio obligatorio y cada criterio deseable se comprueba si se cuenta con la información y estudios clínicos necesarios para evaluarlo.
- Se obtiene una lista fs con los nombres de los estudios faltantes en cada caso:
  - Se agregan todos los elementos de fs a la lista estudiosFaltantes.
  - Si alguna lista fs está vacía, el criterio correspondiente **se puede evaluar**.

## 4. Evaluación de criterios obligatorios

- Para cada criterio obligatorio, **si no se puede evaluar** porque tiene estudios faltantes (resultado del paso anterior), se pasa al paso 7.
- Si el criterio **se puede evaluar**, se procede a evaluar el criterio obteniendo como resultado #apto o #noApto.
  - Si el resultado es #apto, se continúa con el siguiente criterio.
  - Si el resultado es #noApto, se pasa al paso 7.

## 5. Evaluación de criterios deseables

- Se inicializan acumulador y sumaPonderaciones en 0.
- Para cada criterio deseable, se incrementa sumaPonderaciones con la ponderación asociada al criterio.
- Si el criterio **se puede evaluar**, se procede a evaluar el criterio obteniendo como resultado un valor, que se acumula del siguiente modo:
  - $$\text{acumulador} = \text{acumulador} + (\text{ponderación} * \text{valor} / 100).$$

## 6. Terminación con compatibilidad calculada

- En este punto se cuenta con acumulador y sumaPonderaciones válidos para calcular un porcentaje de compatibilidad.

- Se retorna una nueva instancia de ResultadoCompatibilidad con:
  - El paciente y donante considerados.
  - porcentajeCompatibilidad  
= (acumulador / sumaPonderaciones) \* 100
  - estudiosFaltantes

7. Terminación con incompatibilidad absoluta (porcentaje = 0).

- Se retorna una nueva instancia de ResultadoCompatibilidad con:
  - El paciente y donante considerados.
  - porcentajeCompatibilidad = 0.
  - estudiosFaltantes

## Implementación – Clases y métodos solicitados

Se le solicita definir las clases CentroTrasplantes, Persona, EstudioClinico (y subclases), Criterio, CriterioObligatorio (y subclases), CriterioDeseable (y subclases), EvaluadorCompatibilidad y ResultadoCompatibilidad. A continuación se presentan los principales métodos de clase y de instancia solicitados. Para implementar estas funcionalidades puede agregar otras clases o métodos que identifique o requiera.

Clase CentroTrasplantes	
conNombre: nombreDelCentro	<i>Método de clase.</i> Crea y retorna una nueva instancia de CentroTrasplantes con el nombre indicado. Inicializa el estado del objeto como corresponda.
agregaPaciente: unaPersona fechalIngreso: unaFecha	Incorpora la persona recibida como argumento a la lista de pacientes. Se registra unaFecha como la fecha de ingreso del paciente a la lista de espera. No podrán registrarse en el centro a dos personas con el mismo DNI, sean pacientes o donantes.
agregaDonante: unaPersona	Incorpora la persona recibida como argumento a la lista de donantes. No podrán registrarse en el centro a dos personas con el mismo DNI, sean pacientes o donantes.
asociaPaciente: unaPersona conDonante: otraPersona	Asocia entre sí a las instancias de Persona pasadas como argumentos. Establece que el donante indicado es el “donante registrado” del paciente indicado: este donante es un familiar o allegado dispuesto a donar un

	<p>riñón, ya sea al paciente directamente o a un tercero (siempre y cuando el paciente reciba, a su vez, un riñón).</p> <p>Produce un error si alguno de los argumentos no pertenece a la lista de pacientes/donantes ya registrados.</p>
evaluadorCompatibilidad: unEvaluadorCompatibilidad	<p>Define la instancia de EvaluadorCompatibilidad a utilizar. Inicializa o resetea el registro de los resultados de compatibilidad calculados, descartando cualquier resultado que haya sido calculado previamente.</p>
compatibilidad: unPaciente con: unDonante	<p>Calcula la compatibilidad (una instancia de ResultadoCompatibilidad) entre el paciente y donante recibidos, utilizando el objeto EvaluadorCompatibilidad indicado con el método anterior.</p> <p>Si no se definió ningún evaluador, produce un error. Si existe el evaluador, obtiene un objeto ResultadoCompatibilidad para el par (paciente, donante), registra dicho objeto (es decir, lo guarda en su estado interno) y retorna el mismo.</p> <p><b>Nota:</b> El donante no necesariamente será el donante registrado del paciente.</p>
calculaCompatibilidadesRegistrados	<p>Calcula y registra todas las compatibilidades (instancias de ResultadoCompatibilidad) de cada paciente con su donante registrado.</p>
calculaCompatibilidadesGeneral	<p>Calcula y registra todas las compatibilidades posibles, es decir considerando todos los pacientes con todos los donantes.</p>
guardaRegistros: prefijoNombreArchivo	<p>Genera 3 archivos de texto con toda la información de los pacientes, donantes registrados y resultados de compatibilidad. El formato de los archivos será CSV (comma separated values). Ver sección <b>Archivos de salida</b>.</p>

#### Clase Persona

conDni: unDni nombre: unNombre  
apellido: unApellido fechaNacimiento:  
unaFecha

*Método de clase.* Crea y retorna una nueva instancia con los atributos indicados.

✓	peso: unPesoEnKg	Actualiza el peso del receptor.	✓
✓	estatura: unaEstaturaEnMetros	Actualiza la estatura del receptor.	✓
✓	registraEstudio: unEstudioClinico	Asocia el estudio clínico pasado como argumento al receptor, incorporándolo a su lista de estudios clínicos.	
✓	registraEstudio: unEstudioClinico con: unaPersona	Ídem anterior, pero se trata de un estudio clínico realizado para dos personas: el receptor y el argumento. Este método se utilizará para los estudios clínicos del tipo Crossmatch, donde el estudio es específico para el par (paciente, donante).	✓

Clase EvaluadorCompatibilidad			
✓	agregaCriterioObligatorio: unCriterioObligatorio	Agrega el criterio obligatorio pasado como argumento al receptor.	✓
✓	agregaCriterioDeseable: unCriterioDeseable conPonderación: unNumero	Agrega el criterio deseable con la ponderación indicada al receptor.	✓
✓	compatibilidadPaciente: unPaciente con: unDonante	Retorna la compatibilidad entre las instancias de Paciente y Donante recibidas como argumento. Aquí se debe implementar el algoritmo presentado en la sección <b>Evaluación de compatibilidades donante-paciente</b> . El objeto retornado debe ser una instancia de ResultadoCompatibilidad.	

Clase Criterio		
✓	estudiosFaltantes: unPaciente con: unDonante	Método abstracto, debe ser redefinido en las clases concretas. <u>Retorna una OrderedCollection con los nombres (instancias de String) de los estudios faltantes para evaluar el criterio, es decir de aquellos estudios que no fueron realizados.</u>
✓	evaluaCompatibilidad: unPaciente con: unDonante	Método abstracto, debe ser redefinido por las subclases. Los argumentos son instancias de Paciente y Donante, respectivamente. Cada uno de estos

	objetos conoce los estudios clínicos que se le han realizado.
--	---

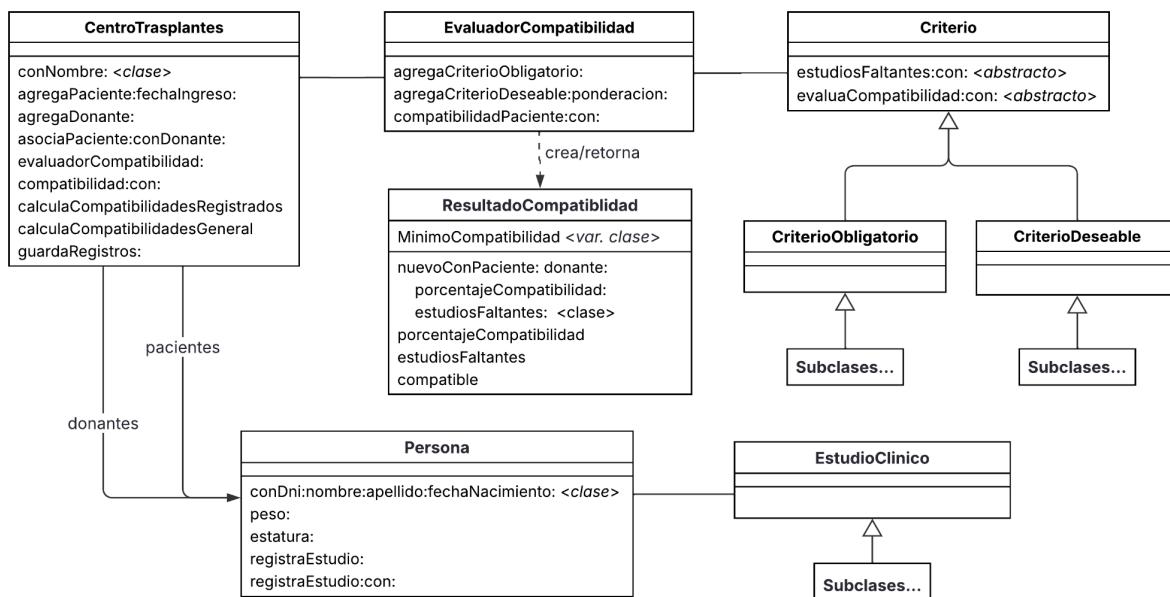
Clase CriterioObligatorio	
evaluaCompatibilidad: unPaciente con: unDonante	<p>Método abstracto, debe ser redefinido por las subclases. Para los criterios obligatorios retorna alguno de los siguientes:</p> <ul style="list-style-type: none"> <li>• #apto (instancia de Symbol): indicando que el criterio se satisface.</li> <li>• #noApto (instancia de Symbol): indicando que el criterio no se satisface.</li> </ul> <p>Si el criterio no puede ser evaluado porque no se realizó alguno de los estudios necesarios, produce un error.</p>

Clase CriterioDeseable	
evaluaCompatibilidad: unPaciente con: unDonante	<p>Método abstracto, debe ser redefinido por las subclases. Para los criterios deseables retorna un número entre 0 y 100, indicando el porcentaje asignado al criterio.</p> <p>Si el criterio no puede ser evaluado porque no se realizó alguno de los estudios necesarios, produce un error.</p>

Clase ResultadoCompatibilidad	
nuevoConPaciente: unPaciente donante: unDonante porcentajeCompatibilidad: unPorcentaje estudiosFaltantes: unaListaNombresDeEstudios	<i>Método de clase.</i> Crea y retorna una nueva instancia del receptor, que contiene la información pasada en los argumentos.
porcentajeCompatibilidad	Retorna el atributo indicado.
estudiosFaltantes	Retorna el atributo indicado.
compatible	Retorna true si el porcentaje de compatibilidad es mayor o igual al mínimo de compatibilidad, caso contrario retorna false.

	<p><u>MinimoCompatibilidad</u> será una variable de clase</p> <p>conteniendo un número entre 0 y 100, que</p> <p>representa el porcentaje mínimo a partir del cual se</p> <p>considera que paciente y donante son compatibles</p> <p>(por ejemplo, 50%).</p>
--	--

## Modelo Orientado a Objetos – Propuesta inicial



## Archivos de salida

Dado un string *prefijoNombreArchivo* se solicita generar tres archivos de datos en formato .csv.

- *prefijoNombreArchivo\_pacientes.csv* incluye una línea para cada paciente, con los siguientes datos (separados por comas):
  - dni,
  - nombre,
  - apellido,
  - cantidad de meses en lista de espera
- *prefijoNombreArchivo\_donantes.csv* incluye una línea para cada donante que tiene un paciente registrado, con los datos siguientes:
  - dni del donante,
  - nombre,
  - apellido,
  - dni del paciente asociado

- ✓
- `prefijoNombreArchivo_compatibilidades.csv` incluye una línea para cada `ResultadoCompatibilidad rc` calculado, tal que `rc compatible` retorne true, con los datos siguientes:
    - dni del paciente,
    - dni del donante,
    - porcentaje compatibilidad

## Ejemplo Playground

```
| centro paciente donante evaluador |

"=====
"CREAR EL CENTRO DE TRASPLANTES"
"=====
centro := CentroTrasplantes conNombre: 'Hospital Central'.
"El centro es el objeto principal que maneja pacientes, donantes y evalúa la
compatibilidad"

"=====
"REGISTRAR PERSONAS"
"=====
"Crear un paciente"
paciente := Persona conDni: 12345671 nombre: 'Juan' apellido: 'Pérez'
            fechaNacimiento: (Date year: 2004 month: 5 day: 23).

"Crear un donante"
donante := Persona conDni: 12345672 nombre: 'María' apellido: 'González'
            fechaNacimiento: (Date year: 1998 month: 3 day: 20).

"Asociar el paciente con su donante registrado (familiar/allegado)"
centro agregaPaciente: paciente.
centro agregaDonante: donante.
centro asociaPaciente: paciente conDonante: donante.

"=====
"CREAR Y REGISTRAR ESTUDIOS CLÍNICOS"
"=====
"Análisis de sangre para determinar grupo sanguíneo"
paciente registraEstudio: (AnalisisDeSangre conGrupoSanguineo: #A). ✓
donante registraEstudio: (AnalisisDeSangre conGrupoSanguineo: #O). ✓

"Crossmatch: estudio específico entre paciente y donante"
paciente registraEstudio: (EstudioCrossmatch conResultado: #negativo) con: donante.

"Serología: detecta infecciones"
paciente registraEstudio: (EstudioSerologicoCompleto conResultado: #apto).
donante registraEstudio: (EstudioSerologicoCompleto conResultado: #apto).

"Tipificación HLA: mide la compatibilidad"
paciente registraEstudio: (EstudioTipificacionHLA conA: #(A1 A2) b: #(B7 B44) dR: #(DR1, DR4)).
donante registraEstudio: (EstudioTipificacionHLA
                           conA: #(A2 A3) b: #(B7 B8) dR: #(DR4, DR15)).
```

```

"Datos de estatura y talla: tamaño corporal"
paciente estatura: 1.75; peso: 74.
donante estatura: 1.80; peso: 82.

"=====
"CREAR CRITERIOS Y CONFIGURAR EL EVALUADOR"
"=====
evaluador := EvaluadorCompatibilidad new.

"Agregar criterios obligatorios (si fallan, no se puede hacer el trasplante)"
evaluador agregaCriterioObligatorio: CriterioABO new.
evaluador agregaCriterioObligatorio: CriterioCrossmatch new.
evaluador agregaCriterioObligatorio: CriterioSerologias new.

"Agregar criterios deseables con sus ponderaciones (peso en %)"
evaluador agregaCriterioDeseable: CriterioHLA new ponderacion: 80.
evaluador agregaCriterioDeseable: CriterioEdad new ponderacion: 10.
evaluador agregaCriterioDeseable: CriterioTamañoCorporal new ponderacion: 10.

"Asignar el evaluador al centro"
centro evaluadorCompatibilidad: evaluador.

"=====
"CÁLCULOS Y GUARDAR ARCHIVOS"
"=====

resultado := centro compatibilidad: paciente con: donante.
"un ResultadoCompatibilidad con porcentajeCompatibilidad = 56.8546"

"Mostrar el resultado"
Transcript show: 'Resultado de compatibilidad: ', show: resultado; cr.

"=====
"CÁLCULOS Y GUARDAR ARCHIVOS"
"=====

"Calcular todas las compatibilidades registradas"
centro calculaCompatibilidadesRegistrados.
Transcript show: 'Compatibilidades registradas calculadas.', cr.

"Calcular todas las compatibilidades posibles"
centro calculaCompatibilidadesGeneral.
Transcript show: 'Compatibilidades generales calculadas.', cr.

centro guardaRegistros: 'ejemploHospitalCentral'.
Transcript show: 'Archivos .csv generados.', cr.

```