DISEÑO DE CASOS DE PRUEBAS

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Prueba No. 1 | Objetivo de la prueba: El objetivo de esta prueba consiste en emparejar a 2 personas siendo estas una mujer (la mujer dependerá de su orientación sexual) y otro usuario. Ademas puede arrojar este tipo de excepciones : PopularException, SexualidadException, SolitarioException, NullPointerException | | | |
| Clase | Método | Escenario | Valores de entrada | Resultado |
| Mujer | Emparejar (Usuario u) | Escenario:  Se va a inicializar un hombre hetero y una pareja hombre hetero | Un usuario con la siguiente descripción:  laMujer = **new** Mujer('A', 'B', 'C', 'D', 'A', 'A', 'C', 'C', 'D', 'A', "Luis", "Luis", Mujer.***HETERO***, "Medicina", 1.59, 20, 4);  Hombre laPareja = **new** Hombre('A', 'B', 'A', 'A', 'A', 'A', 'C', 'C', 'D', 'A', "Cristiano", "Ronaldo", Hombre.***HETERO***, "Telemática", 1.8, 32, 6); | assertEquals()  verificar la pareja de la mujer es igual al hombre que se creó. |
| Mujer | Emparejar (Usuario u) | Escenario:  Se va a inicializar una nueva mujer que va a ser homosexual | Un usuario con la siguiente descripción:  laMujer = **new** Mujer('A', 'B', 'C', 'D', 'A', 'A', 'C', 'C', 'D', 'A', "Luisa", "Luisa", Mujer.BI, "Medicina", 1.59, 20, 4);  Hombre laPareja = **new** Hombre('A', 'B', 'A', 'A', 'A', 'A', 'C', 'C', 'D', 'A', "Cristiano", "Ronaldo", Hombre.***HETERO***, "Telemática", 1.8, 32, 6); | *assertNull*();  por medio del assert null se le va apreguntar a la mujer si tiene pareja pero como la mujer es homosexual retornara null cuando intente encontrar compatibilidad entre la mujer y el hombre |
| Mujer | Emparejar (Usuario u) | Escenario:  Se va a inicializar una nueva mujer bisexual y con nivel de popularidad de 8 | Un usuario con la siguiente descripción:  laMujer = **new** Mujer('A', 'B', 'C', 'D', 'A', 'A', 'C', 'C', 'D', 'A', "Luisa", "Luisa", Mujer.BI, "Medicina", 1.59, 20, 4);  Mujer laPareja = **new** Mujer('A', 'B', 'A', 'A', 'A', 'A', 'C', 'C', 'D', 'A', "Marta", "Ramirez", Mujer.***HOMO***, "Telemática", 1.7, 20, 6); | assertNul()  como la mujer es demasiado popular no se le va asignar pareja sin importar que estén afines según su orientación sexual. Retorna popularException |
| Mujer | Emparejar (Usuario u) | Escenario:  Se va a inicializar una nueva mujer bisexual | Un usuario con la siguiente descripción:  laMujer = **new** Mujer('A', 'B', 'C', 'D', 'A', 'A', 'C', 'C', 'D', 'A', "Luisa", "Luisa", Mujer.BI, "Medicina", 1.59, 20, 4);  Hermafrodita laPareja = **new** Hermafrodita('A', 'C', 'A', 'A', 'C', 'A', 'C', 'C', 'C', 'C', "Alberto", "Ramirez", Mujer.***HOMO***, "Administración", 1.7, 20, 6); | assertNull() a pesar de que ambos usuarios son compatibles sexualmente, la diferencia de las respuestas no los deja ser pareja por lo tanto la pareja es null. Retorna SolitarioException |
|  |  |  |  |  |
|  |  |  |  |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Prueba No. 1 | Objetivo de la prueba: El objetivo de esta prueba consiste en emparejar a 2 personas siendo estas un hombre (el hombre dependerá de su orientación sexual) y otro usuario. Ademas puede arrojar este tipo de excepciones: PopularException, SexualidadException, SolitarioException, NullPointerException | | | |
| Clase | Método | Escenario | Valores de entrada | Resultado |
| Hombre | Emparejar (Usuario u) | Escenario:  Se va a inicializar una nueva mujer que va a ser heterosexual | Un usuario con la siguiente descripción:  elHombre = **new** Hombre('A', 'B', 'C', 'D', 'A', 'A', 'C', 'C', 'D', 'A', "Luis", "Luis", Hombre.***HETERO***, "Medicina", 1.59, 20, 4);  Hombre laPareja = **new** Hombre('A', 'B', 'A', 'A', 'A', 'A', 'C', 'C', 'D', 'A', "Cristiano", "Ronaldo", Hombre.***HETERO***, "Telemática", 1.8, 32, 6); | assertNull()  verificar la pareja del hombre pero como ambos son hetero la pareja ideal es null |
| Hombre | Emparejar (Usuario u) | Escenario:  Se va a inicializar un nuevo hombre que va a ser homosexual | Un usuario con la siguiente descripción:  elHombre = **new** Hombre('A', 'B', 'C', 'D', 'A', 'A', 'C', 'C', 'D', 'A', "Luis", "Luis", Hombre.HOMO, "Medicina", 1.59, 20, 4);  Hombre laPareja = **new** Hombre('A', 'B', 'A', 'A', 'A', 'A', 'C', 'C', 'D', 'A', "Cristiano", "Ronaldo", Hombre.***HETERO***, "Telemática", 1.8, 32, 6); | *assertNull*();  por medio del assertnull se le va encontrar compatibilidad entre un hombre homo y un hombre hetero, pero como no son compatibles retornara la pareja null |
| Hombre | Emparejar (Usuario u) | Escenario:  Se va a inicializar un nuevo hombre hetero y con nivel de popularidad de 8 | Un usuario con la siguiente descripción:  elHombre = **new** Hombre('A', 'B', 'C', 'D', 'A', 'A', 'C', 'C', 'D', 'A', "Luis", "Luis", Hombre.***HETERO***, "Medicina", 1.59, 20, 4);  Mujer laPareja = **new** Mujer('A', 'B', 'A', 'A', 'A', 'A', 'C', 'C', 'D', 'A', "Marta", "Ramirez", Mujer.***HOMO***, "Telemática", 1.7, 20, 6); | assertNul()  como el hombre es demasiado popular no se le va asignar pareja sin importar que estén afines según su orientación sexual.  Retorna excepción PopularException |
| Hombre | Emparejar (Usuario u) | Escenario:  Se va a inicializar un nuevo hombre hetero | Un usuario con la siguiente descripción:  elHombre = **new** Hombre('A', 'B', 'C', 'D', 'A', 'A', 'C', 'C', 'D', 'A', "Luis", "Luis", Hombre.***HETERO***, "Medicina", 1.59, 20, 4);  Hermafrodita laPareja = **new** Hermafrodita('A', 'C', 'A', 'A', 'C', 'A', 'C', 'C', 'C', 'C', "Alberto", "Ramirez", Mujer.***HOMO***, "Administración", 1.7, 20, 6); | assertNull()  debido a que los usuarios son diferentes en su orientación retornara la excepción SolitarioExcepcion |
|  |  |  |  |  |
|  |  |  |  |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Prueba No 1 | Objetivo de la prueba: El objetivo de esta prueba consiste en emparejar a 2 personas siendo estas un hermafrodita (el hermafrodita dependerá de su orientación sexual) y otro usuario. Ademas puede arrojar este tipo de excepciones: PopularException, SexualidadException, SolitarioException, NullPointerException | | | |
| clase | Método | escenario | Valores de entrada | Resultado |
| Hermafrodita | Emparejar(Usuario u) | Escenario :  Crea un nuevo hermafrodita homosexual y un hombre hetero | Un usuario con la siguiente descripción:  persona = **new** Hermafrodita('A', 'B', 'C', 'D', 'A', 'A', 'C', 'C', 'D', 'A', "Luis", "Luis", Hermafrodita.***HOMO***, "Medicina", 1.59, 20, 4);  Hombre laPareja = **new** Hombre('A', 'B', 'A', 'A', 'A', 'A', 'C', 'C', 'D', 'A', "Cristiano", "Ronaldo", Hombre.***HETERO***, "Telemática", 1.8, 32, 6); | assertNull()  como existe una diferencia en la sexualidad del hermafrodita arroja SexualidadException y retorna null en la pareja ideal. |
| Hermafrodita | Emparejar(Usuario u) | Escenario :  Crea un nuevo hermafrodita Bi y un hombre hetero | Un usuario con la siguiente descripción:  persona = **new** Hermafrodita('A', 'B', 'C', 'D', 'A', 'A', 'C', 'C', 'D', 'A', "Luis", "Luis", Hermafrodita.***BI***, "Medicina", 1.59, 20, 4);  laPareja = **new** Hombre('A', 'B', 'A', 'A', 'A', 'A', 'C', 'C', 'D', 'A', "Cristiano", "Ronaldo", Hombre.***HETERO***, "Telemática", 1.8, 32, 6); | assertNull()  como la orientación sexual es diferente retorna la pareja del hermafrodita null |
| Hermafrodita | Emparejar(Usuario u) | Escenario :  Crea un nuevo hermafrodita homo pero el hermafrodita tiene popularidad de 8 y una mujer homo | Un usuario con la siguiente descripción:  persona = **new** Hermafrodita('A', 'B', 'C', 'D', 'A', 'A', 'C', 'C', 'D', 'A', "Luis", "Luis", Hermafrodita.***HOMO***, "Medicina", 1.59, 20, 4);  Hombre  Mujer laPareja = **new** Mujer('A', 'B', 'A', 'A', 'A', 'A', 'C', 'C', 'D', 'A', "Marta", "Ramirez", Mujer.***HOMO***, "Telemática", 1.7, 20, 6); | assertNull()  como el Hermafrodita es demasiado popular no importa si tienen afinidad no serán pareja por lo tanto retorna null y se activa la excepción PopularException |
| Hermafrodita | Emparejar(Usuario u) | Escenario :  Crea un nuevo hermafrodita homo y otro hermafrodita homo | Un usuario con la siguiente descripción:  persona = **new** Hermafrodita('A', 'B', 'C', 'D', 'A', 'A', 'C', 'C', 'D', 'A', "Luis", "Luis", Hermafrodita.***HOMO***, "Medicina", 1.59, 20, 4);  Hermafrodita laPareja = **new** Hermafrodita('A', 'C', 'A', 'A', 'C', 'A', 'C', 'C', 'C', 'C', "Alberto", "Ramirez", Mujer.***HOMO***, "Administración", 1.7, 20, 6); | assertNull()  como la orientación sexual de ellos es diferente entonces retorna null y se activa la excepción SolitarioExcepcion |
|  |  |  |  |  |
|  |  |  |  |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Prueba No 1 | Objetivo de la prueba: El objetivo de esta prueba es ver si el método coincidencias esta funcionando correctamente, el método hacer match, el método contar, el método registrar usuario, el método agregar usuario, el metedo resetear.  Ademas puede arrojar este tipo de excepciones en el metodod hacer match: PopularException, SexualidadException, SolitarioException, NullPointerException | | | |
| clase | Método | escenario | Valores de entrada | Resultado |
| IcesiMatch | generaraCoincidencias() | Escenario:  Crea 32usuarios nuevos que serán agregados al árbol | **new** Hombre('A', 'B', 'A', 'C', 'D', 'A', 'A', 'A','D', 'D', "James", "James", Usuario.***HETERO***, "Ingeniería Telemática", 1.85, 22, 6))  **new** Hombre('A', 'B', 'A', 'C', 'D', 'A', 'A', 'A','D', 'D', "Falcao", "Falcao", Usuario.***HOMO***, "Medicina", 2, 19, 4))  **new** Mujer('D', 'C', 'A', 'A', 'C', 'A', 'C', 'A','C', 'D', "Daniela", "Daniela", Usuario.***HETERO***, "Quimica", 1.57, 20, 2)); | assertEquals()  como las repuestas de los usuarios son iguales el número de coincidencias es 10 por lo tanto son iguales |
| IcesiMatch | Contar() | Escenario :  Crea 2 usuarios nuevos que serán agregados al árbol | **new** Hombre('A', 'B', 'A', 'C', 'D', 'A', 'A', 'A','D', 'D', "James", "James", Usuario.***HETERO***, "Ingeniería Telemática", 1.85, 22, 6))  **new** Hombre('A', 'B', 'A', 'C', 'D', 'A', 'A', 'A','D', 'D', "Falcao", "Falcao", Usuario.***HOMO***, "Medicina", 2, 19, 4))  **new** Mujer('D', 'C', 'A', 'A', 'C', 'A', 'C', 'A','C', 'D', "Daniela", "Daniela", Usuario.***HETERO***, "Quimica", 1.57, 20, 2)); | assertEquals () el valor esperado es 2 y el valor retornado es 2 porque solo hay 2 usuarios agregados |
| IcesiMatch | hacerMatch(); | Escenario :  Crea 2 nuevos usuarios con iguales caracteristicas | **new** Hombre('A', 'B', 'A', 'C', 'D', 'A', 'A', 'A','D', 'D', "James", "James", Usuario.***HETERO***, "Ingeniería Telemática", 1.85, 22, 6))  **new** Hombre('A', 'B', 'A', 'C', 'D', 'A', 'A', 'A','D', 'D', "Falcao", "Falcao", Usuario.***HOMO***, "Medicina", 2, 19, 4))  **new** Mujer('D', 'C', 'A', 'A', 'C', 'A', 'C', 'A','C', 'D', "Daniela", "Daniela", Usuario.***HETERO***, "Quimica", 1.57, 20, 2)); | assertNull()  Intentara buscar su pareja ideas pero como ambos usuarios no son compatibles retornara la pareja null y se activara el Solitario Exception |
| IcesiMatch | registrarUsuario(Usuario u) | Escenario :  Crea 2 nuevos usuarios con iguales características | Se crea un usuario con la siguiente información : ("hola","hola", "hola", "Homo Sexual", "hola", 0, 0,0); | assertEquals() al agregar el usuario correctamente le pedimos el usuario actual al modelo y debe de ser igual al usuario que se agrego |
| IcesiMatch | AgregarActual(); | Escenario:  Creo 2 usuarios con las mismas caracteristicas | Se crea un usuario con la siguiente información :  Hombre('A', 'B', 'A', 'C', 'D', 'A', 'A', 'A','D', 'D', "James", "James", Usuario.***HETERO***, "Ingeniería Telemática", 1.85, 22, 6))  **new** Hombre('A', 'B', 'A', 'C', 'D', 'A', 'A', 'A','D', 'D', "Falcao", "Falcao", Usuario.***HOMO***, "Medicina", 2, 19, 4))  **new** Mujer('D', 'C', 'A', 'A', 'C', 'A', 'C', 'A','C', 'D', "Daniela", "Daniela", Usuario.***HETERO***, "Quimica", 1.57, 20, 2)); | AssertEquals() como agrego el mismo usuario actual entonces la cantidad de usuarios debe de ser 3 |
| IcesiMatch | resetear(); | Escenario :  Se agregaron 2 usuarios iniciales | Se crea un usuario con la siguiente información :  Hombre('A', 'B', 'A', 'C', 'D', 'A', 'A', 'A','D', 'D', "James", "James", Usuario.***HETERO***, "Ingeniería Telemática", 1.85, 22, 6))  **new** Hombre('A', 'B', 'A', 'C', 'D', 'A', 'A', 'A','D', 'D', "Falcao", "Falcao", Usuario.***HOMO***, "Medicina", 2, 19, 4))  **new** Mujer('D', 'C', 'A', 'A', 'C', 'A', 'C', 'A','C', 'D', "Daniela", "Daniela", Usuario.***HETERO***, "Quimica", 1.57, 20, 2)); | assertEquals()  como se resetearon el numero de coincidencias entonces no importa cuanta afinidad tenga el resultado será 0 |