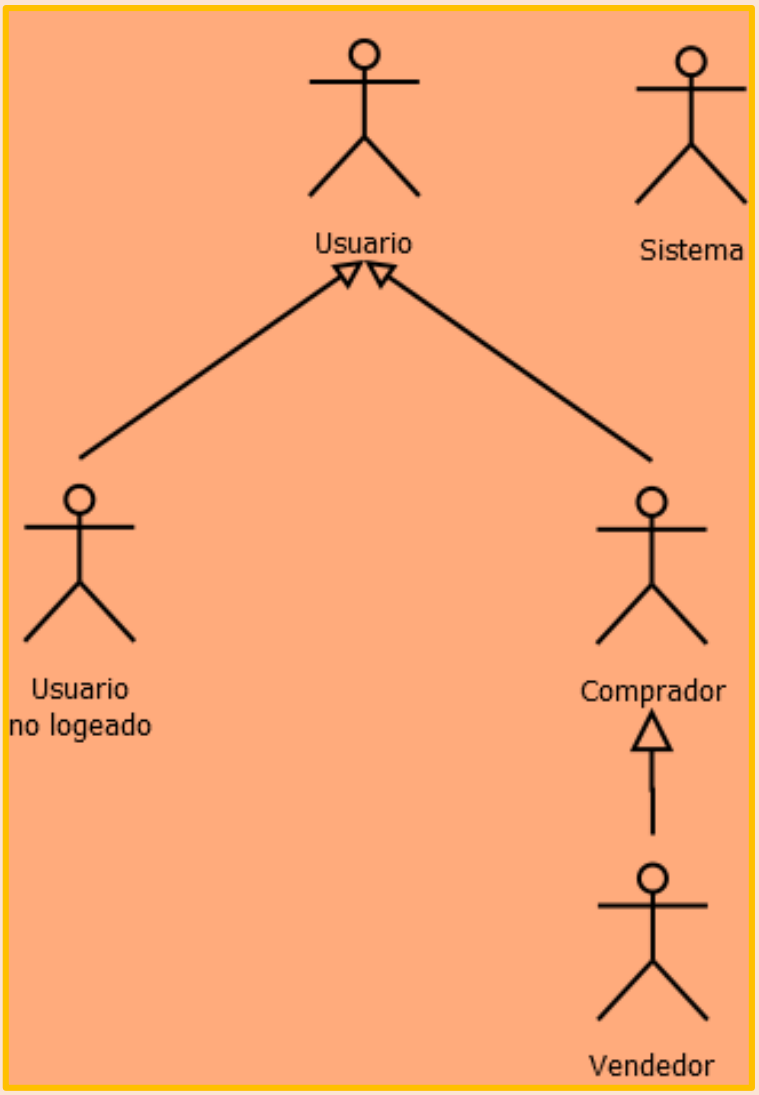
Sitio Web de Subastas

Laura Calvente Domínguez

INTRODUCCIÓN

En este proyecto encontraremos principalmente cinco entidades:

- **Usuario:** Cualquier persona puede realizar las actividades que corresponden a este perfil.
- **Usuario no logueado:** Persona que además de realizar las acciones de cualquier usuario puede registrarse o iniciar sesión.
- **Comprador:** Puede realizar las actividades del usuario además de las suyas propias
- **Vendedor:** hará lo mismo que el comprador además de subastar productos.



DOCUMENTACIÓN

Usuario

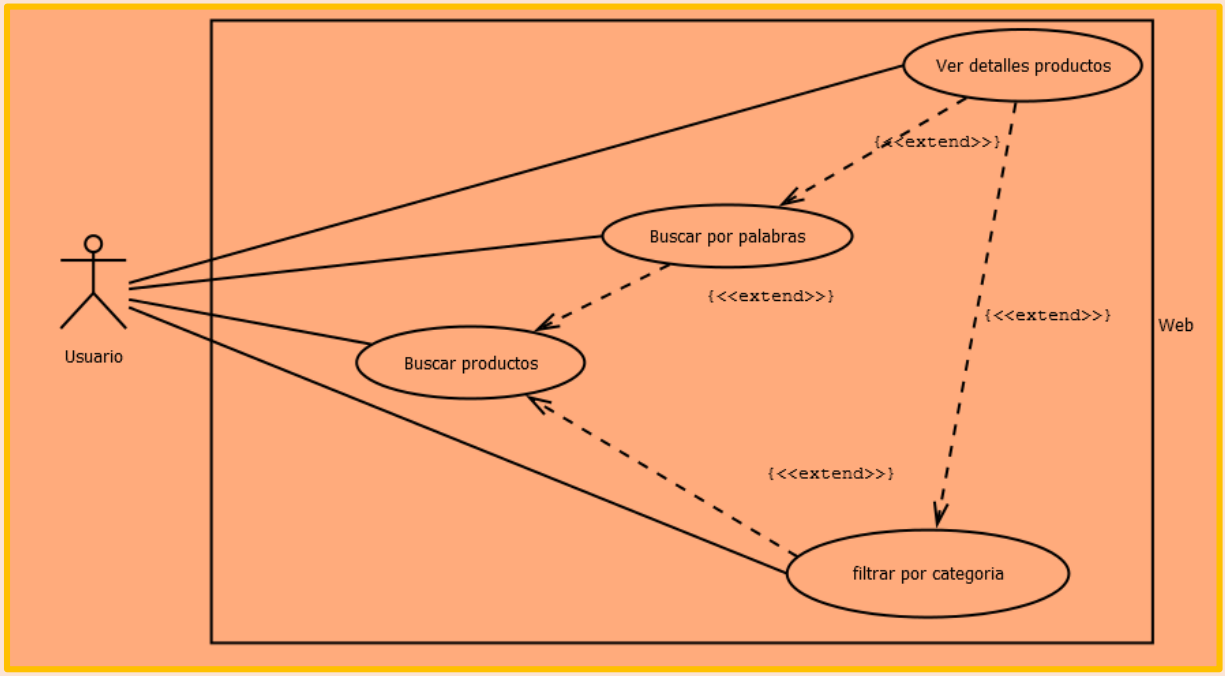
En nuestro modelo el **usuario** no es un actor de la vida real, su única función será dar sentido a la estructura hereditaria del proyecto. Este se encargara de realizar las acciones de búsqueda. Usuario tiene dos hijos que son: el usuario no logueado y el comprador.

- **Buscar Productos**: Cualquier usuario (logueado o no) podrá buscar productos. Cuando se realiza una búsqueda sin especificar palabras ni categoría se mostraran todos los productos subastados ordenados ascendentemente por tiempo de finalización.

- **Buscar por palabras**: Se podran realizar busquedas a través de palabras clave. es decir, las palabras clave tienen que estar todas contenidas en el nombre del producto como palabras o parte de palabras, sin distinguir entre mayusculas y minusculas, y en cualquier orden.

- **Filtrar por categorías**: La busqueda puede ser filtrada por categorias y subcategorias. El resultado de esta se limitara a los productos de la categoría en cuestión.

- **Ver detalles productos**: Una vez realizada una busqueda podemos acceder a los detalles de un producto concreto.



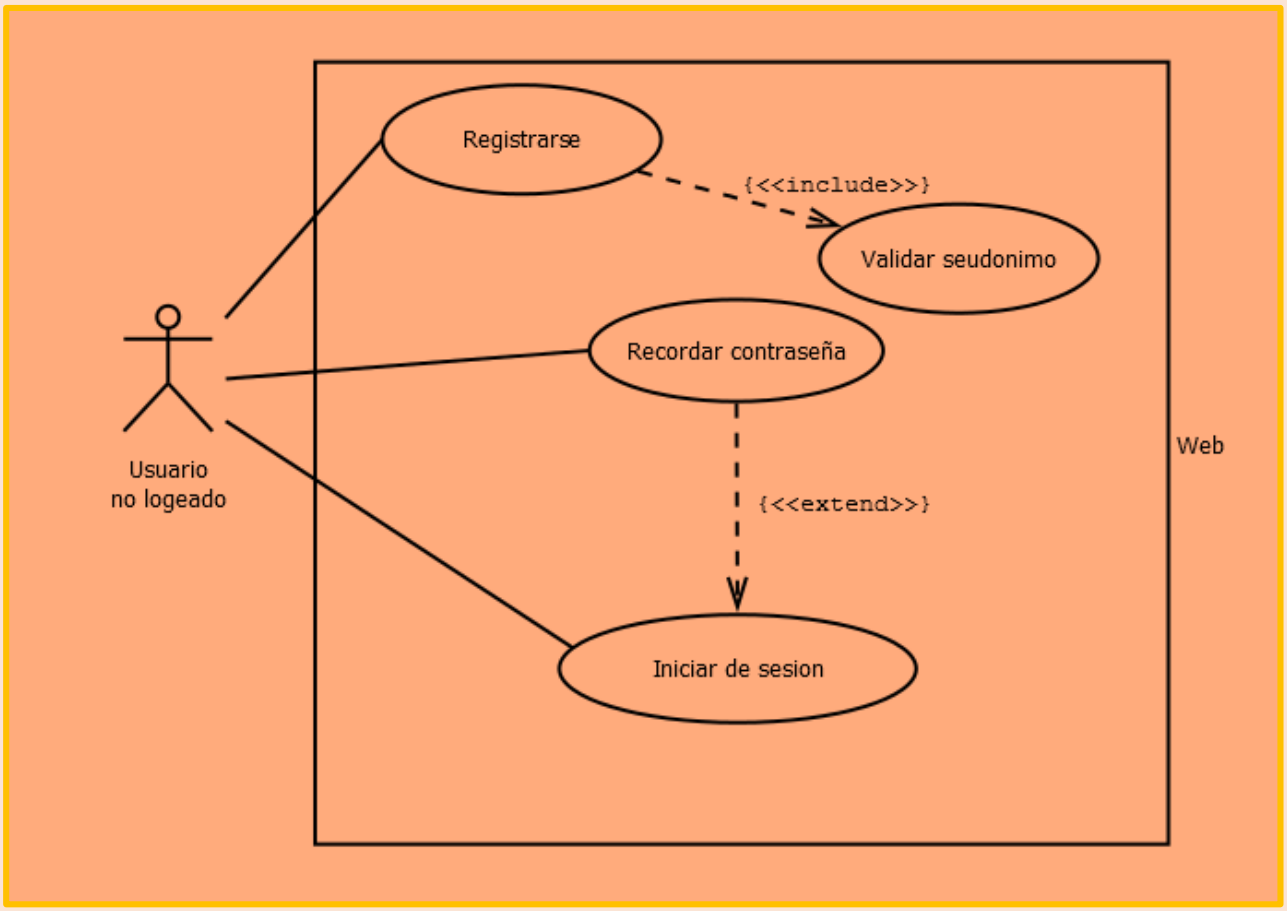
Usuario no logueado

Usuario no logueado heredara las acciones de su padre usuario, además de tener las suyas propias. Este rol será asignado a aquel que entre en la web sin haberse logueado.

- **Registrarse**: Un usuario al registrarse no puede introducir directamente los datos bancarios, si este quiere convertirse en vendedor lo deberá hacer una vez iniciada la sesión tras registrarse. El controlador internamente debe comprobar que el seudónimo no exista ya en la base de datos además de cumplir con un patrón.

- **Iniciar sesión**: Un usuario podrá iniciar sesión con su seudónimo y su contraseña.

- **Recordar contraseña**: Cuando un usuario se loguea puede recordar la contraseña.



Comprador

El **comprador** es un usuario logueado que implementara las acciones del usuario que es su padre además de las suyas propias. Este tiene un hijo vendedor.

- **Cerrar sesión**: Un usuario con sesión iniciada siempre podrá cerrar su sesión, pasando a ser un usuario no logueado. Esta acción incluye olvidar la contraseña.

- **Realizar pedido**: esta opción se dará a la hora de mostrar los detalles de un producto. El controlador debe comprobar que la puja está bien realizada.

- **Actualizar datos del registro**: un usuario logueado sea vendedor o comprador, podrá modificar los datos de su registro. Para esto pide de nuevo los datos de iniciar sesión para verificar que es la misma persona.

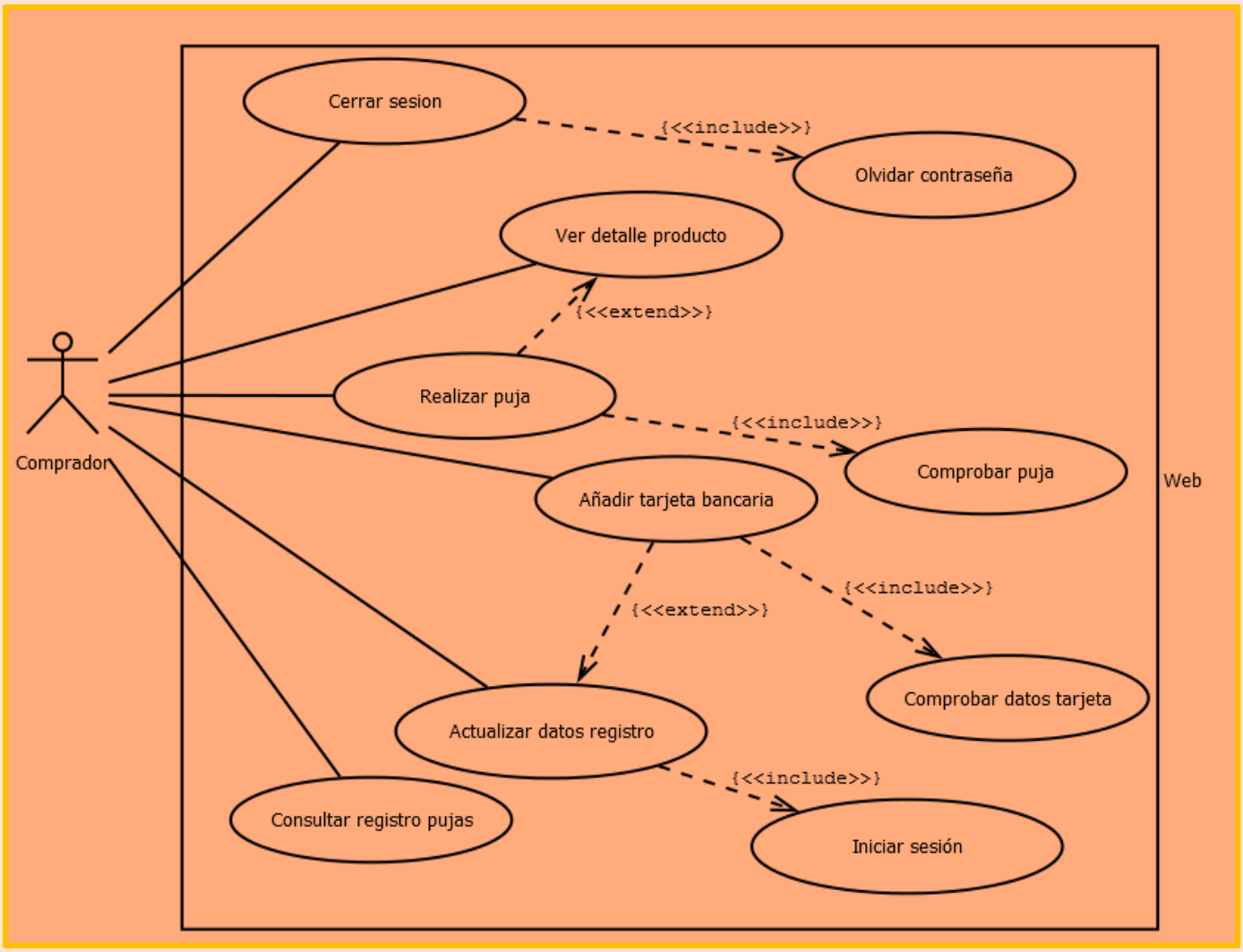
  Nota programador: Puntuamos aquí que en el método del "CompradorFacadeDelegate", "modificarRegristro (String campo, String dato)", el parámetro campo que le llega seria el objeto capturado por el ratón con un script y el dato sería la nueva información a introducir. De esta forma cuando un usuario modifica su información de registro no estará obligado a volver a rellenarlo todo.

  En este apartado también podemos añadir o modificar la información bancaria.

- **Añadir tarjeta bancaria**: Si un usuario desea ser vendedor tras registrarse y loguearse no le quedara otra opción que cumplimentar los datos bancarios, no obstante si un usuario ya es vendedor, el programa usara el mismo método para cambiar la información bancaria. El controlador debe comprobar que los datos bancarios son correctos.

  Nota programador: El método debe preguntar si es una instancia del "Comprador_J2Facade" o del "Vendedor_J2EFacade". Ya que, este método no se puede redefinir tal y como ha sido concebido el diagrama de clases. Solo habría un action.

- **Consultar registro de pujas**: devuelve una lista ordenada descendientemente por las más recientes.



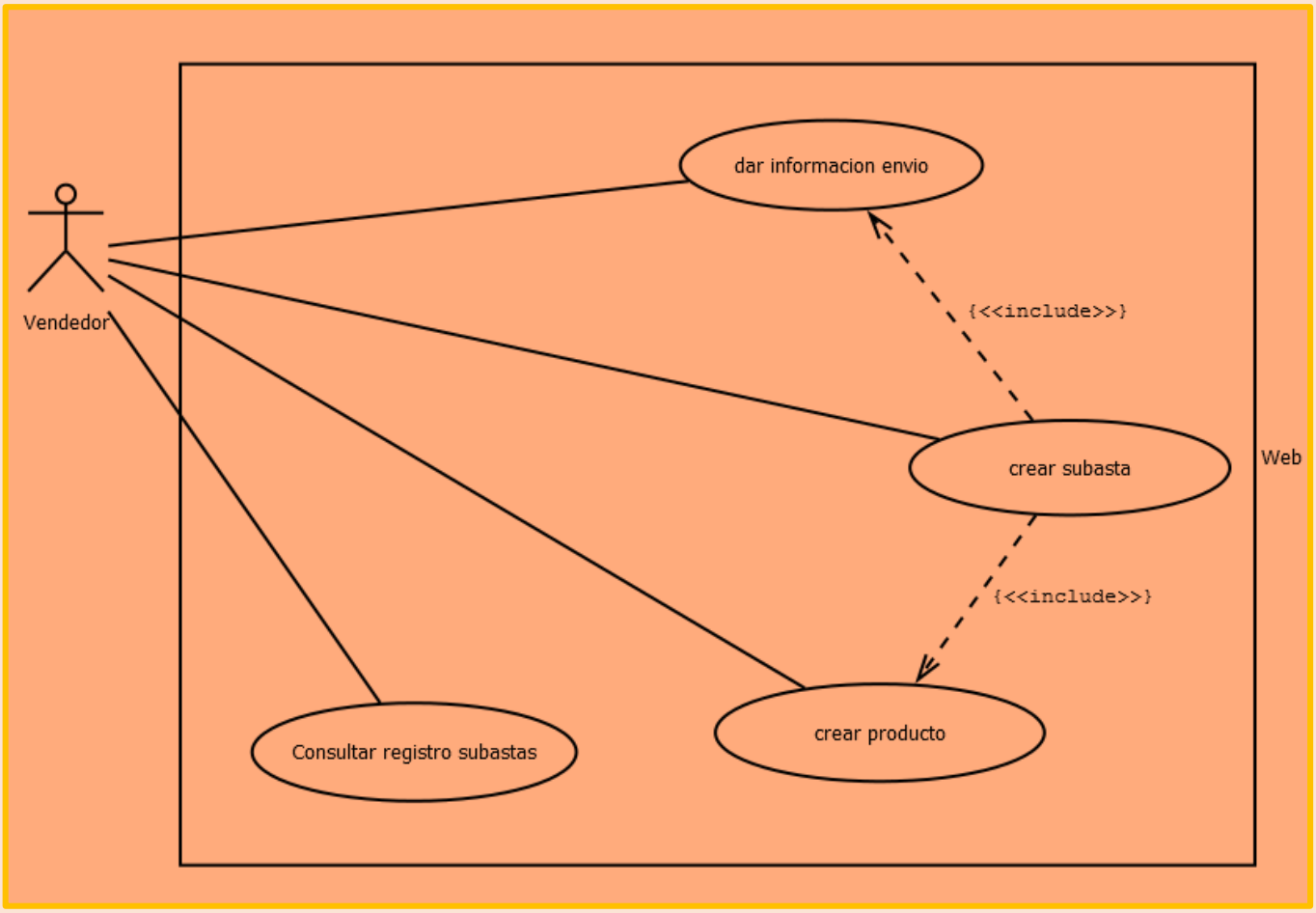
Vendedor

El **vendedor** implementa las acciones del usuario ya que las hereda por su padre comprador además de las de este y las suyas propias.

- **Crear subasta**: Un vendedor tendrá opción a anunciar una nueva subasta, con un nuevo producto y una nueva información de envió.
- **Crear producto**: cuando se crea una subasta se deba crear antes el producto con todos sus datos correspondientes.
- **Dar información de envió**: Al crear una subasta se dará la información de envió. Esta información será almacenada en una tabla independiente en la base de datos y en el

modelo quedara asociado siempre a la subasta. De esta forma un vendedor podría cambiar la información de envió de una subasta.

- **Consultar registro subastas**: Al igual que se puede mirar el registro de pujas, un vendedor también pobra acceder al registro de sus subastas.



Sistema

El **sistema** es un actor ajeno a la herencia descrita anteriormente que se encargara de ejecutar sus métodos constantemente. Podemos decir que es quien refresca los datos.

- **Actualizar puja**: método que no devolverá nada. Su función es llamar a los demás métodos que el sistema debe ejecutar constantemente, independientemente por ejemplo del comprobar tarjeta bancaria, o comprobar puja que se encargara el controlador.

- **Adjudicar puja**: si el tiempo restante de la subasta ha expirado y había un comprador vencedor se le adjudicara la subasta a ese comprador. El estado de la subasta cambiaria (boolean activo = false).

- **Cerrar puja**: si el tiempo restante para la expiración de la puja ha transcurrido y no ha habido ningún comprador la puja se finalizara sin tener un comprador. El estado de la subasta cambiaria (boolean activo = false).

- **Actualizar tiempo puja**: se encarga de actualizar constantemente el tiempo que debe mostrar cada subasta.

- **Incrementar puja**: busca todas las subastas activas de la base de datos y va comprobando constantemente si ha habido un incremento en la subasta y entonces sube la puja de cada comprador, teniendo en cuenta que no debe superar el máximo marcado por el comprador en esa puja.