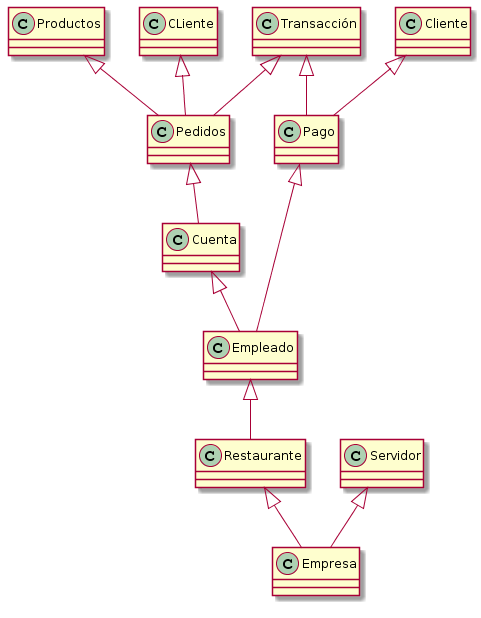
# 5. Proyecto de implementación de un prototipo del sistema utilizando PlantUML

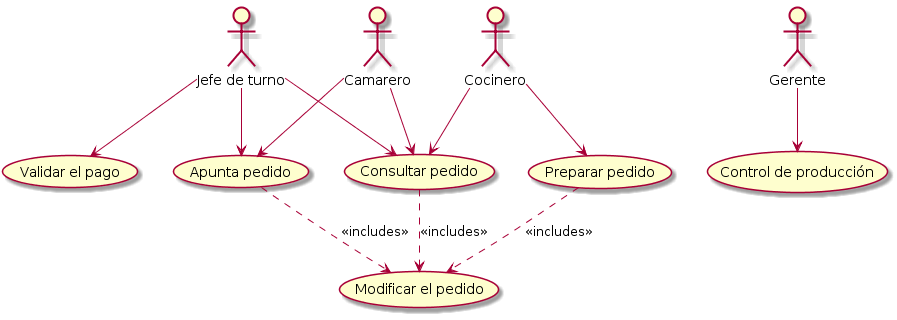
## 5.1 Documentación de diseño

En este caso utilizaremos el mismo prototipo que hemos utilizado en la herramienta anterior, y realizaremos los tres diagramas para poder comparar entre ambas herramientas:

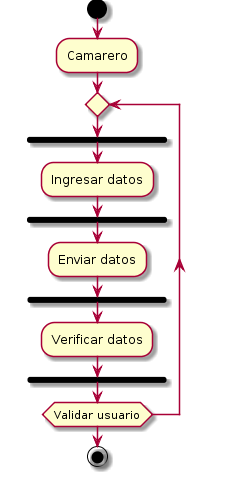
* Diagrama de clases:



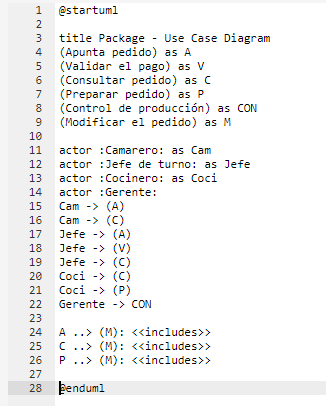
* Diagrama de caso de uso:



* Diagrama de actividades:



## 5.2 Documentación de construcción

El código de esta herramienta varía un poco de lo conocido habitualmente, pero acaba siendo intuitivo con la ayuda de todos los ejemplos de diagramas que incluye la herramienta. Para cada diagrama nos aporta distintos ejemplos de las cosas básicas que podríamos necesitar en cuanto a código. Por ejemplo, para realizar el diagrama de casos de uso nos ofrece ejemplos de casos de uso básicos, especificación de actores, conexiones, estereotipos o direcciones, lo que ayuda mucho a la hora de saber como hacer el diagrama sin tener muchos conocimientos de la herramienta. Por otra parte, en esta herramienta, para poder visualizar el diagrama primero hay que tener escrito el código, y una vez escrito procederemos a darle al botón “refresh”, el cual hace que aparezca el diagrama.

En la siguiente imagen observamos el código del diagrama de casos de uso en el que primero asignamos los nombres acortados para que sea más sencillo nombrarlos, después, declaramos los actores, los cuales podemos también asignarle otro nombre o acortarlo. Por último, trazamos las relaciones con flechas simples o con dos puntos si queremos hacer línea no continua.

## 5.3 Documentación de pruebas

Como casos de prueba hemos establecido un prototipo de software de otra asignatura y hemos intentado rehacer el caso y los diagramas mediante la herramienta online de PlantUML, que es PlantText: <https://www.planttext.com/>

En la realización de los distintos diagramas han ido surgiendo varios problemas, como:

* Al haber errores de código (sobre todo al inicio de la implementación debido al poco conocimiento sobre el lenguaje del código) simplemente muestra un mensaje indicando que hay un error, concretando la primera línea de código en la que muestra el error y lo que hay escrito en dicha línea, pero no indica cuál es exactamente el error, o no ayuda a saber como solucionarlo. Además, al haber un error no muestra ningún diagrama, y como el diagrama no se carga hasta que no pulsamos el botón “refresh”, no podemos saber si hay errores en el código hasta entonces.
* También hemos encontrado dificultades a la hora de orientar los diagramas ya que influye mucho el orden del código a la hora de la colocación de los actores. Al principio nos aparecía un actor en cada esquina del diagrama y líneas cruzadas por todo el diagrama con las relaciones. Esto supone un lío a la hora de interpretar el diagrama ya que no sabíamos si lo estábamos haciendo correctamente.
* Al realizar el diagrama de actividades, hemos encontrado más complicaciones. No nos aparecía el diagrama, pero tampoco nos indicaba que hubiera ningún error por lo que no sabíamos si lo estábamos haciendo bien, o era problema de la herramienta por que no estaba mostrando el diagrama.

## 5.4 Documentación de instalación

Esta herramienta nos permite dos opciones: acceder de manera online, o descargar la extensión de PlantUML para instalarla en otras herramientas más conocidas.

Nosotros hemos optado por la opción online, ya que este trabajo se basa en comparar dos herramientas de modelado UML online, por lo que los pasos para acceder a la herramienta de PlantUML online son sencillos. Como parte fundamental necesitamos tener acceso a internet para acceder a la página de <https://www.planttext.com/> en la que podremos comenzar a utilizar la herramienta onlie.

## 5.5 Manual de usuario

En primer lugar identificamos la herramienta accediendo a la página web de [PlantText](https://www.planttext.com/) (RF01), una vez aquí nos muestra una pantalla de código para poder escribir los diagramas, y encima de la pantalla un menú desplegable en el cual nos da todas las posibilidades de diagramas de PlantUML (RF02). En el menú desplegable también nos indica ayudas para editar los diagramas, o añadir anotaciones (RF05, RF07). Una vez que tenemos escrito el código para visualizar el diagrama debemos dar al botón de “refresh” para que se nos muestre el diagrama en la parte derecha de la pantalla. Para modificar o eliminar el diagrama simplemente deberemos cambiar el código escrito (RF04, RF09, RF10). Si queremos guardar el diagrama nos da varias opciones de formato, simplemente seleccionamos en el que queremos guardarlo (RF06). Otra opción que tenemos es registrarnos, se puede hacer fácilmente con la cuenta de Google, y así podremos guardar nuestros diagramas para modificarlos posteriormente.

# 8. Conclusiones

Bajo nuestro punto de vista, ambas herramientas nos han resultado muy útiles para realizar diagramas UML de manera online. Nos aportan rapidez ya que no es necesario que nos descarguemos e instalemos ningún programa, siempre que tengamos una conexión a internet.

En comparación, la herramienta de yUML resulta más llamativa a simple vista ya que su interfaz es bastante más amigable que la de PlantUML, y además es más sencilla de encontrar la URL de la herramienta online. Cabe destacar la simplicidad de código de yUML frente a PlantUML. El lenguaje que utilizamos en yUML resulta más cercano al lenguaje humano, y con la ayuda de los ejemplos que incluyen es bastante sencillo hacerse a la idea de cómo funciona el código.

Como desventaja de yUML hemos comprobado que, si tenemos algún error en el diagrama, simplemente no se muestra, ni india que exista error alguno, por lo que ralentiza bastante la tarea si no te das cuenta del error. Otra de las desventajas que encontramos en esta herramienta es que, a la hora de visualizar los diagramas se muestran en la parte inferior de la pantalla, por lo que si escribimos demasiado código llega un momento en el que dejamos de ver el diagrama y esto resulta bastante molesto e incómodo a la hora de trabajar con diagramas mas extensos ya que si escribes bastante código sin mirar el diagrama y después lo observas y ha habido un error, no se verá ningún diagrama ni errores, y habrá que borrar la mayor parte y observar dónde nos hemos equivocado.

Por otra parte, analizando la herramienta de PlantUML, el lenguaje que ésta utiliza es más lenguaje tipo código, no es tan cercano al lenguaje humano, ni tan complicado como lenguaje de programación, pero lo suficiente como para tardar un poco más examinando cómo realizar el diagrama. Otro de los motivos que ralentiza la ejecución del diagrama en esta herramienta es que tenemos que esperar a tener escrito el código para después visualizar el diagrama pulsando un botón. Esto resulta bastante incómodo ya que no vas viendo el proceso de tu diagrama y pierdes más tiempo realizando las modificaciones.

Las ventajas de PlantUML es que ofrece muchos ejemplos de diagramas y especifica más ejemplos dentro de cada uno de ellos. Otro de los pros de esta herramienta es que el diagrama se visualiza a la derecha de la pantalla, pudiendo escribir el código y posteriormente visualizarlo completo sin tener que desplazar la pantalla. Y, por último, otra de las ventajas que hemos apreciado en esta herramienta es que, a la hora de identificar errores resulta mucho más sencillo ya que indica en un panel en que línea está el error y cual es el código erróneo.

Como conclusión, si tenemos que usar alguna de estas dos herramientas para un determinado proyecto, en primer lugar, valoraríamos el tamaño del proyecto. Si se trata de un proyecto como el que hemos realizado en esta comparación (sencillo y corto) utilizaríamos sin duda la herramienta A, yUML, ya que nos proporciona más rapidez y facilidad con el diagrama. Sin embargo, si tenemos que realizar un proyecto más complejo o extenso, nos decantamos más por la herramienta B, PlantUML, ya que nos aporta más ayuda en información, ya sea de realización de diagramas o de información sobre los errores.