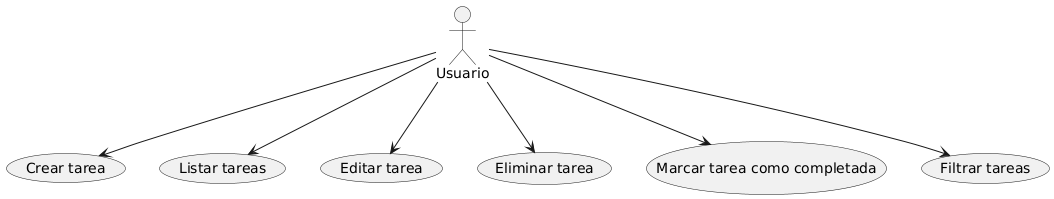
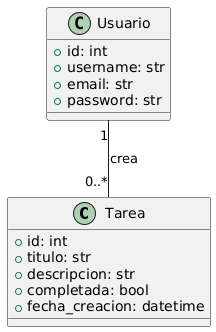
**Trabajo Practico Integrador   
Programación Avanzada  
  
Comisión**: 3  
**Grupo**: 5  
**Integrantes**:  
-Florencia Daniela Pivetta   
-Jesica Carla Lencina   
-Nicolás Alejandro Seculini   
-Matías Ezequiel Gonzalez   
  
 En el presente ensayo, explicaremos como decidimos como grupo la elección del tema “Aplicaciones web con Frameworks Python” y como desarrollamos la aplicación.   
 En primer lugar, como grupo, debatimos si había algún interés particular en alguna de las temáticas propuestas y analizamos que varios consideraron que la realización de una aplicación web era interesante. Una vez elegido el tema, se plantearon las diferentes aplicaciones que podíamos realizar y optamos por la elección de una aplicación web de administración de tareas al estilo “ToDo…”.  
 Una vez resuelta la elección del tema y de la funcionalidad de nuestra aplicación, procedimos a elegir el framework que utilizaríamos para realizarla. Optamos por la utilización de Django, debido a las facilidades que brinda justamente en cuestiones de desarrollo de aplicaciones web. Ademas de ser código abierto y escrito en Python.

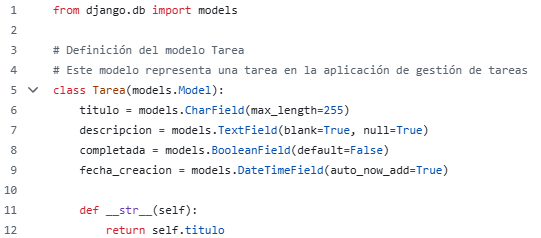
Comenzamos en ese momento un proceso de investigación y aprendizaje sobre como aplicarlo, realizando diferentes tutoriales al respecto.   
 Mientras tanto, se plantearon las diferentes funciones que buscábamos que nuestra aplicación pudiera cumplir, pensando principalmente en las necesidades que tendría un usuario a la hora de utilizarla.



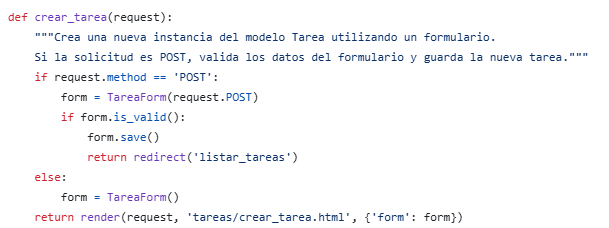
Las principales funciones para nosotros y en las que priorizamos fue en la creación de tareas, que tuvieran tanto un titulo que la identifique, una descripción mas explayada sobre dicha tarea y un estado sobre su estado. A su vez, debíamos permitir la edición o eliminación en caso de errores que existieran a la hora de que un usuario las creara mal. Eso nos permitió entender que, considerando el paradigma de Programación Orientada a Objetos (POO) era fundamental la creación de la clase “Tarea”, cuyo modelo para realizarla es el siguiente:



A partir de ese modelo, lo llevamos al código creando nuestra clase:



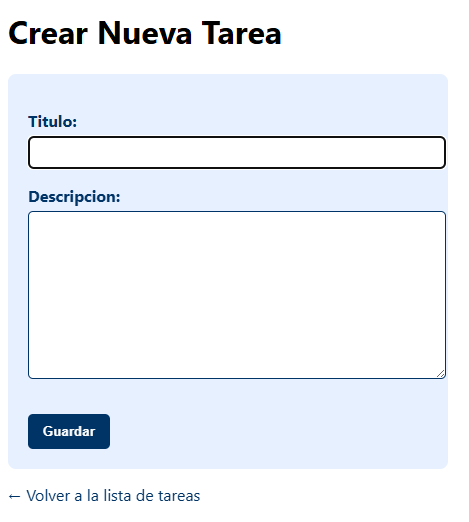
Una vez creada, debíamos crear los diferentes métodos que nos permitieran todas las acciones que nos habíamos propuesto desde un inicio.  
  
 Un ejemplo de estos métodos es el de creación de tareas:



Una vez finalizado todo el proceso de creación del código, la aplicación nos quedo de la siguiente manera. La pagina principal de la aplicación muestra el listado de diferentes tareas creadas donde se visualiza el “Titulo” y el estado de las diferentes actividades, diferenciándose con color verde las “Completadas” y con rojo las “Pendientes”. Debajo del listado, esta el enlace que permite la creación de las nuevas tareas.



Al acceder a “+ Crear nueva tarea”, nos envía a la siguiente pagina, donde podemos ver un formulario que permite agregar el titulo y la descripción de la tarea a crear y, en caso de arrepentirse antes de guardarla, una opción para volver al listado anterior.



Como siguiente punto importante, al entrar en alguna tarea en especifico dentro del listado, accedemos a las opciones de editar, eliminar y cambiar estado de la tarea seleccionada.



Con esto, la aplicación logro funcionar y cumplir con nuestras expectativas. Aun así, consideramos como grupo que existen varias mejoras y actualizaciones que se le podrían hacer. Se planteo las ideas de organizar o filtrar actividades por categorías, entre otras cuestiones visuales y organizativas que facilitarían a usuarios en casos donde existan demasiadas actividades por hacer. Aun asi, el grupo se sintió conforme con el avance que logramos hacer y ver como pudimos aplicar sobretodo lo aprendido a la Programación Orientada a Objetos como paradigma a la hora de programar y desarrollar una aplicación.  
 Creemos que con tiempo, con respecto a las mejoras que mencionamos con anterioridad, podrían añadirse con la creación de diferentes usuarios o categorías, añadiendo ambas cosas como clases, para aplicar métodos que permitirían facilitar mas la relación de los diferentes apartados de la aplicación. Ademas, luego de seguir investigando Django, descubrimos como hacer las vistas basadas en clases (Class-based view) en Django. Así, hubiéramos aplicado de mejor forma el polimorfismo al permitirnos extender y personalizar el comportamiento de las vistas