

Técnicas Avanzadas de Programación

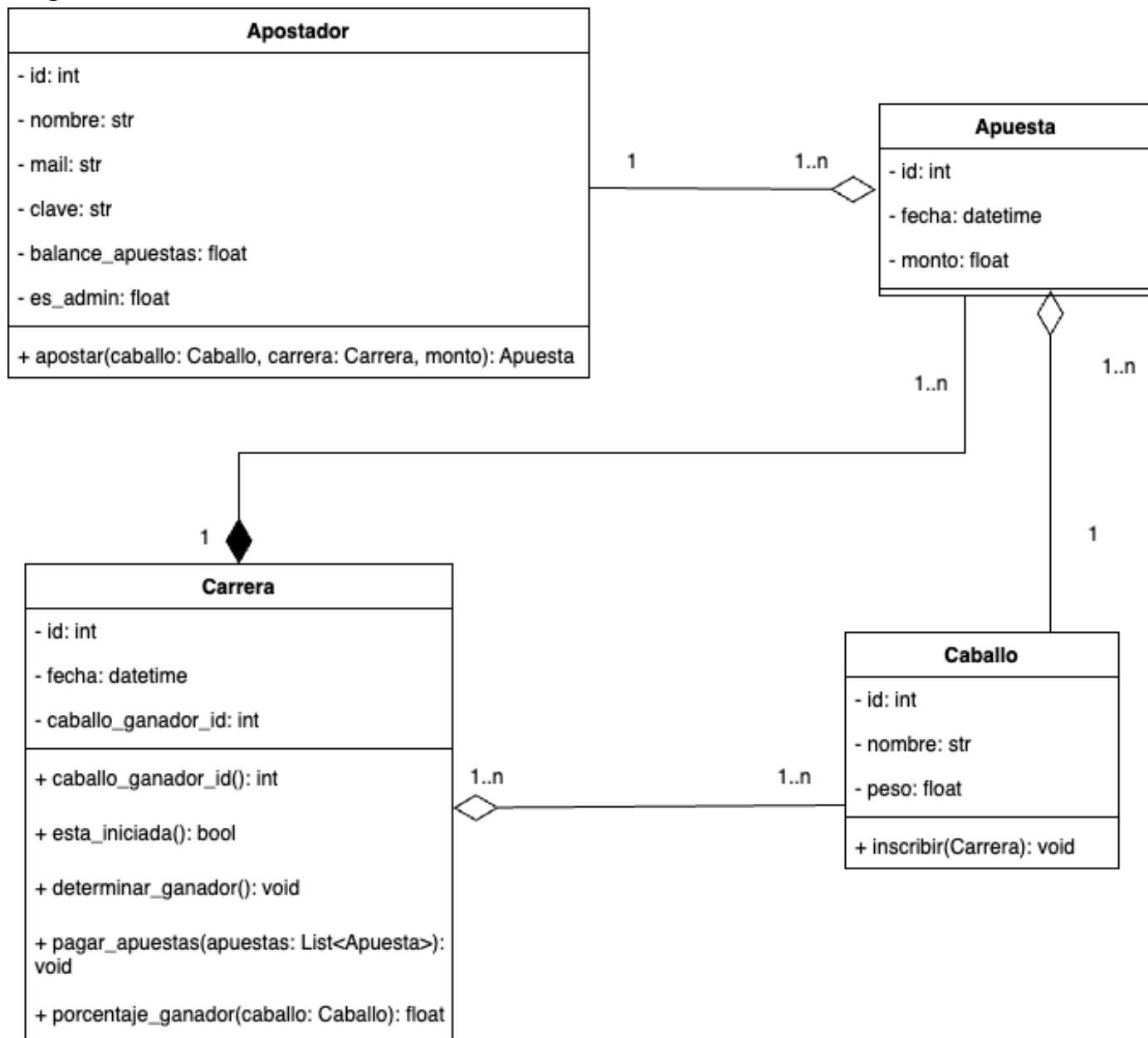
Trabajo práctico integrador - Consigna
primera etapa



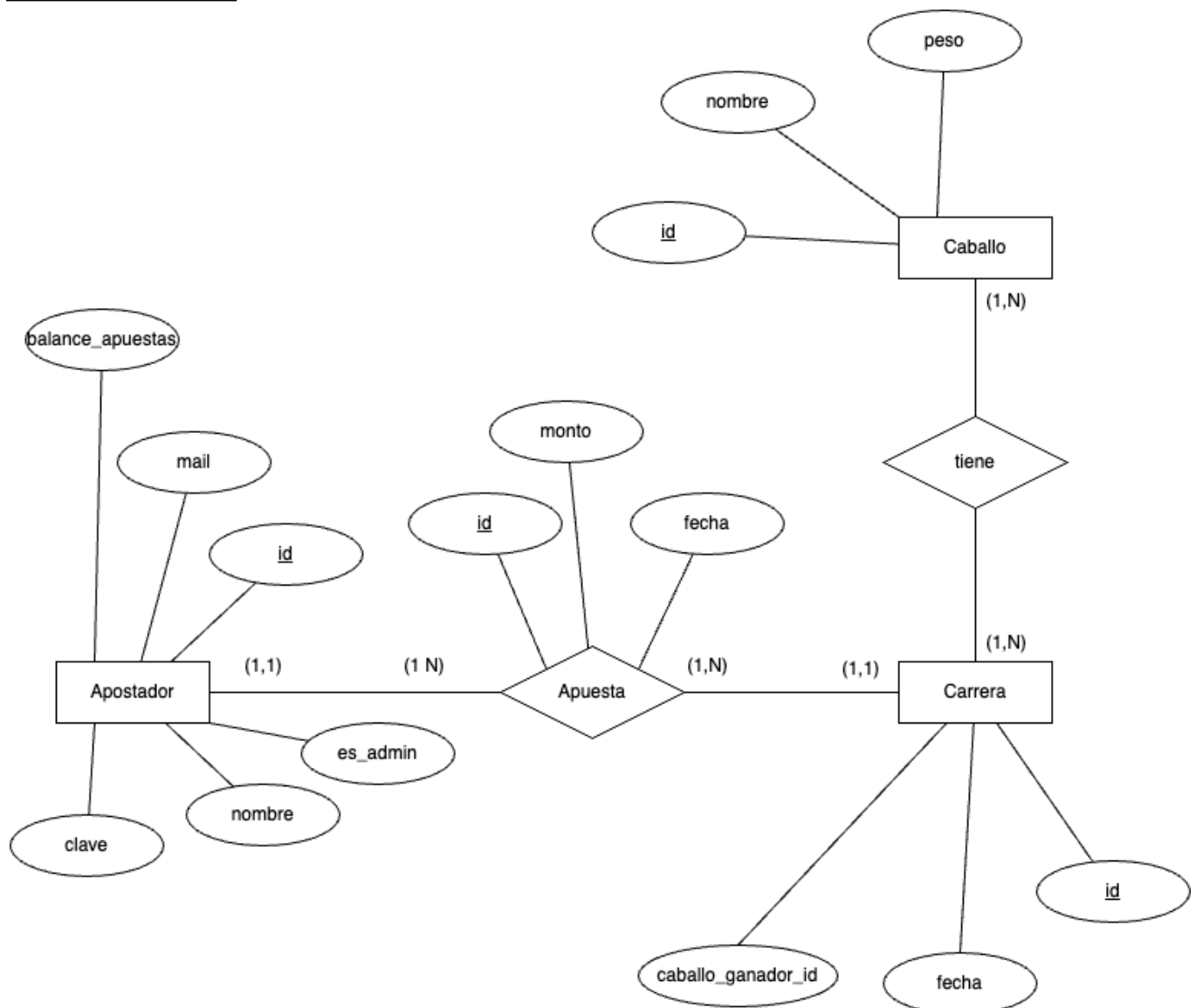
Alumno: Carro, Nahuel Agustín

Año 2024

Diagrama de clases:



Modelo de datos:



Detalle de las tecnologías a utilizar:

Las tecnologías a utilizar son Python, donde se va a utilizar como Framework FastAPI. Para la base de datos del ORM se va a utilizar PostgreSQL y para desplegar todo el trabajo Docker.

Tipo de testeo y módulos que se planifican testear:

Los tests se realizan con la librería pytest.

- Test unitarios:
 - **test_inscribir_caballo:** prueba la inscripción de un caballo a una carrera
 - **test_login_con_contrasena_incorrecta:** prueba un login con contraseña incorrecta
 - **test_ver_perfil_sin_login:** prueba ver el perfil de un usuario sin estar logueado
- Test de integración:
 - **test_apostar_con_monto_menor_a_cero:** se loguea un usuario e intenta crear una apuesta con monto incorrecto

- **test_apostar_con_carrera_iniciada:** se loguea el usuario y se intenta apostar con una carrera ya iniciada
- **test_agregar_caballo_sin_ser_admin:** se intenta agregar un caballo sin ser administrador
- **test_iniciar_carrera_y_no_es_la_fecha:** se intenta iniciar una carrera que todavía no es la fecha de comienzo de la carrera
- **test_listar_carreras:** se realiza un test de listar las carreras disponibles
- **Test del sistema:**
 - **test_del_sistema:** se prueba la creación de un usuario, el login, crear una carrera y caballo, se agrega el caballo a la carrera, se realiza una apuesta y se empieza una carrera verificando que se haya depositado en el campo de balance_apuestas el monto ganado.

Justificación de la elección de tecnología y consideraciones:

Justificación para usar FastAPI: elegí utilizar FastAPI ya que vengo trabajando con Python. Además FastAPI permite crear APIs de manera más sencilla que Django que es otro framework de Python. Utiliza Pydantic que valida automáticamente los datos de entrada y salida basándose en los tipos de Python, permite generar documentación automática con OpenAPI (Swagger) sin configuraciones adicionales, esto facilita el uso de la API para otros desarrolladores. Además soporta asincronía lo que permite manejar operaciones que pueden demorar mucho tiempo y tiene un rendimiento muy alto.

Justificación de usar PostgreSQL: es una base de datos con la que he trabajado y además tiene varias ventajas como su compatibilidad con SQLAlchemy que es el ORM que voy a usar con FastAPI, está diseñado para manejar múltiples operaciones concurrentes, ofrece transacciones ACID lo que asegura que las operaciones sean seguras y consistentes. En cuanto a la escalabilidad, PostgreSQL es altamente escalable, tanto vertical como horizontalmente, lo que permite manejar grandes volúmenes de datos.