

Elemento UML	Restricción OCL	Implementación Python	Prueba
Clase Reserva.estado: EstadoReserva	inv EstadoValido: Set{...}->includes(self.estado)	class EstadoReserva(str, Enum)	Enum) test_crear_reserva_ok
Asociación Tutor[1] ↔ Reserva[*]	inv TieneTutorYEstudiante: not self.tutor.ocllsundefined() inv	reserva.tutor_id: str (required)	(required) test_crear_reserva_ok
Invariante No Duplicados	NoDobleReservaMismoHorario: ...->isUnique(r r.fechaHora)	for r in reservas: if r.fecha_hora == fecha_hora...	test_crear_reserva_falla_doble_reserva_activa
Secuencia "crear reserva"	pre CrearReservaValida: tutor.disponibilidades->exists(...)	disp_repo.find_free(tutor.id, fecha_hora)	test_crear_reserva_falla_sin_disponibilidad
Estado CREADA → CONFIRMADA	Transición válida en máquina de estados	(no implementado en código Python básico)	-
Estado CREADA → CANCELADA	Transición válida en máquina de estados	service.cancelar_reserva()	test_cancelar_reserva_ok
Estado CANCELADA → X	Transición inválida	pre CancelarSoloSiActiva → validación en Python	test_cancelar_reserva_falla_si_no_activa
Postcondición crear	post ReservaQuedaCreada: result.estado = EstadoReserva::CREADA	estado=EstadoReserva.CREADA en constructor	test_crear_reserva_ok