



Temporización

Comienzo de la práctica	Duración
Semana del 19 de mayo	3 semanas

Actividades a realizar

A partir de los contratos proporcionados en este guión se llevarán a cabo las siguientes actividades:

1. Creación del *modelo de interacción de objetos* de las operaciones especificadas, representándolo mediante un conjunto de diagramas de comunicación de UML.
2. Obtención del *diagrama de clases del diseño* a partir del modelo de interacción de objetos obtenido anteriormente, representándolo mediante un diagrama de clases de UML.

Recomendaciones para realizar la práctica

El modelo de interacción de objetos debe hacerse aplicando las reglas explicadas en clase de teoría en el tema 3.3. Este modelo representa el conjunto de objetos que colaboran entre sí para llevar a cabo una operación, junto con las responsabilidades de cada uno de ellos. Para ello, sirven como base los contratos y, como guía, el modelo conceptual.

A continuación, se deberá crear el diagrama de clases del diseño, teniendo en cuenta todo el proceso explicado en el tema 3.4. En este caso, se usará como base el modelo de interacción de objetos y, como guía, el modelo conceptual.

Entrega de los documentos generados

Documentación a aportar (Tipo de entrega)	Fecha de entrega
<ul style="list-style-type: none">• Modelo de interacción de objetos (Individual). Cada estudiante debe diseñar un mínimo de tres operaciones y cada diagrama debe llevar el nombre del estudiante que lo ha realizado• Diagrama de clases del diseño (Grupal)	10-junio-2025

GESTIÓN DE RESERVAS

Nombre	idCliente = altaCliente(dni, nombre, apellidos, email, telefonoMovil, tarjetaPago, usuarioPayPal)
Responsabilidad	Crear un nuevo Cliente
Tipo	GERAT
Notas	
Excepciones	<ul style="list-style-type: none"> dni no está en el formato correcto tarjetaPago no está en el formato correcto usuarioPayPal no está en el formato correcto
Salida	idCliente del objeto Cliente creado
Precondiciones	
Poscondiciones	<ul style="list-style-type: none"> Se creó el objeto Cliente, unCliente, debidamente inicializado y con valor de atributo esUsuario a false

Nombre	idCliente = getIdCliente(dni, nombre, apellidos)
Responsabilidad	Obtener el idCliente correspondiente a un Cliente con los valores dados para dni, nombre, apellidos
Tipo	GERAT
Notas	
Excepciones	<ul style="list-style-type: none"> dni no está en el formato correcto No existe un Cliente con los valores dados de dni, nombre, apellidos
Salida	idCliente del Cliente con los valores dados de dni, nombre, apellidos
Precondiciones	
Poscondiciones	

Nombre	modificarCliente(idCliente, dni, nombre, apellidos, email, telefonoMovil, tarjetaPago)
Responsabilidad	Modificar los datos del Cliente identificado por idCliente
Tipo	GERAT
Notas	
Excepciones	<ul style="list-style-type: none"> dni no está en el formato correcto tarjetaPago no está en el formato correcto
Salida	
Precondiciones	<ul style="list-style-type: none"> El Cliente identificado por idCliente debe existir
Poscondiciones	<ul style="list-style-type: none"> Se modificaron los datos del objeto Cliente identificado por idCliente, siendo los nuevos datos los proporcionados en los parámetros

Nombre	infoCliente=getInfoCliente(idCliente)
Responsabilidad	Obtener y devolver los datos del Cliente identificado por idCliente
Tipo	GERAT
Notas	
Excepciones	
Salida	infoCliente = {dni, nombre, apellidos, email, telefonoMovil, tarjetaPago, usuarioPayPal} para el Cliente identificado por idCliente
Precondiciones	<ul style="list-style-type: none"> El Cliente identificado por idCliente debe existir
Poscondiciones	

Nombre	listaInfoReservas = listarReservas(datosBusqueda)
Responsabilidad	Obtener una lista de todas las reservas que verifiquen los datos de búsqueda dados
Tipo	GERAT
Notas	
Excepciones	
Salida	listaInfoReservas = lista de {infoReserva} donde infoReserva = {idCliente} del objeto Cliente que realiza la Reserva + {idEvento} del objeto Evento para el que se ha realizado la Reserva + {idCompra, fecha} del objeto Compra realizada por el Cliente + {idParticipante, accesibilidad, rango_edad} del objeto Participante para el que se ha realizado la reserva
Precondiciones	
Poscondiciones	

Nombre	listaInfoReservas = listarReservas(idCliente)
Responsabilidad	Obtener una lista de todas las reservas realizadas por el Cliente identificado por idCliente
Tipo	GERAT
Notas	
Excepciones	
Salida	listaInfoReservas = {idCliente} + lista de {infoReserva} donde infoReserva = {idEvento} del objeto Evento para el que se ha realizado la Reserva + {idCompra, fecha} del objeto Compra realizada por el Cliente + {idParticipante, accesibilidad, rango_edad} del objeto Participante para el que se ha realizado la Reserva
Precondiciones	<ul style="list-style-type: none"> El Cliente identificado por idCliente debe existir
Poscondiciones	

Nombre	precio, idPago = cancelarReserva(idCliente, idReserva)
Responsabilidad	Cancelar la Reserva realizada por un Cliente para un determinado Evento
Tipo	GERAT
Notas	
Excepciones	
Salida	<ul style="list-style-type: none"> - precio del objeto Reserva identificado por idReserva - idPago del objeto Pago asociado al objeto Compra en el que está incluido el objeto Reserva identificado por idReserva y adquirido por el objeto Cliente identificado por idCliente
Precondiciones	<ul style="list-style-type: none"> ▪ El Cliente identificado por idCliente debe existir ▪ La Reserva identificada por idReserva debe existir
Poscondiciones	<ul style="list-style-type: none"> ▪ Se eliminó el enlace del objeto Reserva identificado por idReserva y el objeto Evento para el que se había realizado la Reserva ▪ Se eliminó el enlace del objeto Reserva identificado por idReserva y el objeto Participante para el que se había realizado la Reserva

Nombre	comprobanteReembolso = reembolsoReserva(idCliente, idReserva, idPago, precio)
Responsabilidad	Obtener la devolución de la cantidad pagada por una reserva
Tipo	GERAT
Notas	
Excepciones	<ul style="list-style-type: none"> ▪ precio distinto al que aparece en el objeto Reserva identificado por idReserva
Salida	Comprobante de que ha sido devuelto el precio de la reserva
Precondiciones	<ul style="list-style-type: none"> ▪ El Cliente identificado por idCliente debe existir ▪ La Reserva identificada por idReserva debe existir ▪ El Pago identificado por idPago debe existir
Poscondiciones	<ul style="list-style-type: none"> ▪ Se creó el objeto de la clase Reembolso, unReembolso, debidamente inicializado a la fecha y hora actual ▪ Se creó un enlace entre el objeto unReembolso y el objeto Pago identificado por idPago ▪ Se modificó el dato cantidad del objeto Pago identificado por idPago a cantidad-precio ▪ Se eliminó el enlace “incluye” del objeto Reserva identificado por idReserva y el objeto Compra adquirido por el objeto Cliente identificado por idCliente ▪ Se eliminó el objeto Reserva identificado por idReserva

Nombre	altaClienteRegistrado(idCliente, nombreUsuario, password)
Responsabilidad	Dar de alta a un Cliente como cliente registrado, asignándole un nombre de usuario y una contraseña
Tipo	GERAT
Notas	
Excepciones	<ul style="list-style-type: none"> Existe otro cliente registrado con el mismo nombreUsuario password no está en el formato correcto
Salida	
Precondiciones	<ul style="list-style-type: none"> El Cliente identificado por idCliente debe existir
Poscondiciones	<ul style="list-style-type: none"> Se creó un objeto de la clase Usuario, unUsuario, debidamente inicializado Se creó un enlace entre unUsuario y el objeto Cliente identificado por idCliente

Nombre	idCompra, idReserva, precio = iniciarCompra (idCliente, idParticipante, accesibilidad, idEvento,)
Responsabilidad	Iniciar la compra de entradas (reservas) para un evento de una actividad
Tipo	GERAT
Notas	
Excepciones	
Salida	idCompra del objeto Compra creado idReserva del objeto Reserva creado precio de la Reserva creada y calculado como se indica en las poscondiciones
Precondiciones	<ul style="list-style-type: none"> El Cliente identificado por idCliente debe existir El Evento identificado por idEvento debe existir El Participante identificado por idParticipante debe existir
Poscondiciones	<ul style="list-style-type: none"> Se creó un objeto de la clase Compra, unaCompra, debidamente inicializado a la fecha actual Se creó un enlace entre unaCompra y el objeto Cliente identificado por idCliente. Se creó un objeto de la clase Reserva, unaReserva, debidamente inicializado, con precio calculado a partir del rango_edad del objeto Participante identificado por idParticipante, y de las ofertas que tenga asociadas el objeto de la clase Evento identificado por idEvento Se creó un enlace entre el objeto unaReserva y el objeto Evento identificado por idEvento Se creó un enlace entre objeto Participante identificado por idParticipante y el objeto Evento identificado por idEvento Se creó un enlace entre el objeto Evento identificado por idEvento y el objeto unaCompra

Nombre	idParticipante = altaParticipante (nombre, apellidos, dni, rango_edad)
Responsabilidad	Dar de alta a un participante de un evento
Tipo	GERAT
Notas	
Excepciones	<ul style="list-style-type: none"> • dni no está en el formato correcto • rango_edad no tiene un valor válido
Salida	idParticipante del objeto participante creado
Precondiciones	
Poscondiciones	<ul style="list-style-type: none"> • Se creó el objeto de la clase Participante, unParticipante, debidamente inicializado

Nombre	idReserva, precio = reservarEvento(idCompra, idParticipante, accesibilidad)
Responsabilidad	Hacer la reserva de un Evento para un Participante
Tipo	GERAT
Notas	
Excepciones	<ul style="list-style-type: none"> • Se ha excedido el número de plazas disponibles del Evento • Se ha excedido el número máximo de reservas por cliente para ese Evento
Salida	<ul style="list-style-type: none"> - idReserva del objeto Reserva creado - precio de la Reserva creada y calculado como se indica en las poscondiciones
Precondiciones	<ul style="list-style-type: none"> • La Compra identificada por idCompra debe existir • El Participante identificado por idParticipante debe existir
Poscondiciones	<ul style="list-style-type: none"> • Se creó un objeto de la clase Reserva, unaReserva, debidamente inicializado, con precio calculado a partir del valor rango_edad del objeto Participante identificado por idParticipante, y de las ofertas que tenga asociadas el objeto de la clase Evento con el que está enlazada la Compra identificada por idCompra • Se creó un enlace entre unaReserva y el objeto Participante identificado por idParticipante • Se creó un enlace entre unaReserva y el objeto Evento con el que está enlazada la Compra identificada por idCompra • Se creó un enlace entre unaReserva y el objeto Compra identificado por idCompra

Nombre	idPago, pagado = pagoCompra(idCompra, tipo_Pago, cantidad)
Responsabilidad	Hacer el Pago correspondiente a una(s) Reserva(s)
Tipo	GERAT
Notas	
Excepciones	<ul style="list-style-type: none"> • tipo_Pago no tiene un valor válido • cantidad es distinta a la suma de todos los precios correspondientes a las reservas incluidas en la Compra identificada por idCompra
Salida	El idPago del objeto Pago creado pagado = valor booleano que si está a true significado que el pago se ha realizado
Precondiciones	<ul style="list-style-type: none"> • El objeto Compra identificado por idCompra debe existir
Poscondiciones	<ul style="list-style-type: none"> • Se creó un objeto de la clase Pago, unPago, debidamente inicializado • Se creó un enlace entre unPago y el objeto Compra identificado por idCompra

Nombre	Rollback(idCompra)
Responsabilidad	Deshacer una compra de entradas (reservas)
Tipo	GERAT
Notas	
Excepciones	
Salida	
Precondiciones	<ul style="list-style-type: none"> • El objeto Compra identificado por idCompra debe existir
Poscondiciones	<ul style="list-style-type: none"> • Para todas las Reservas incluidas en la Compra identificada por idCompra <ul style="list-style-type: none"> - Se eliminó el enlace entre la Reserva y el Evento para el que se había realizado Reserva - Se eliminó el enlace entre la Reserva y el Participante para el se había realizado la reserva - Se eliminó el enlace entre la Reserva y la Compra identificada por idCompra - Se eliminó la Reserva • Se eliminó el enlace entre la Compra identificada por idCompra y el Cliente que realizó la Compra • Se eliminó el objeto Compra identificado por idCompra

GESTIÓN DE ACTIVIDADES

Nombre	idActividad = definirActividad(tipo_actividad, infoActividad)
Responsabilidad	Definir una nueva Actividad
Tipo	GERAT
Notas	
Excepciones	<ul style="list-style-type: none"> tipo_Actividad no tiene un valor válido
Salida	idActividad: identificador del objeto de la clase Actividad creado
Precondiciones	
Poscondiciones	<ul style="list-style-type: none"> Se creó un objeto de la clase Actividad debidamente inicializado

Nombre	idEvento = definirEvento(idActividad, infoEvento)
Responsabilidad	Definir un nuevo evento asociado a una actividad identificada por idActividad
Tipo	GERAT
Notas	
Excepciones	<ul style="list-style-type: none"> plazas_disponibles mayor que el valor del atributo máximo_asistencia que aparece en el objeto de la clase Actividad identificado por idActividad cancelado igual a true finalizado igual a true inicio de periodo_cancelacion posterior a la fecha del evento fin de periodo_cancelacion anterior al inicio de periodo_cancelacion fin de periodo_cancelacion anterior a la fecha actual
Salida	idEvento: identificador del objeto de la clase Evento creado
Precondiciones	<ul style="list-style-type: none"> El objeto de la clase Actividad identificado por idActividad debe existir
Poscondiciones	<ul style="list-style-type: none"> Se creó un objeto de la clase Evento, unEvento, debidamente inicializado. Al atributo máximo_reservas_cliente se le asignó el valor 3 (por defecto) Se creó un enlace entre el objeto unEvento y el objeto de la clase Actividad identificado por idActividad

Nombre	modificarActividad(idActividad, infoActividad)
Responsabilidad	Modificar los datos de la Actividad identificada por idActividad
Tipo	GERAT
Notas	
Excepciones	
Salida	
Precondiciones	<ul style="list-style-type: none"> El objeto de la clase Actividad identificado por idActividad debe existir
Poscondiciones	<ul style="list-style-type: none"> Se modificaron los datos del objeto de la clase Actividad identificado por idActividad, siendo los nuevos datos los proporcionados en los parámetros

Nombre	infoActividades=listarActividades(criterio_búsqueda)
Responsabilidad	Obtener una lista de todas las actividades que verifiquen el criterio de búsqueda dado
Tipo	GERAT
Notas	
Excepciones	
Salida	infoActividades = lista de infoActividad donde infoActividad = {nombre, duracion, transporte, comida, guia, audioguia, idioma, maximo_asistencia, precio} para aquellas actividades que verifiquen el criterio_búsqueda dado
Precondiciones	
Poscondiciones	

Nombre	activarPeriodoReserva(idEvento, fechaInicio, fechaFin)
Responsabilidad	Especificar el periodo de reserva de un evento por parte del encargado
Tipo	GERAT
Notas	
Excepciones	<ul style="list-style-type: none"> ▪ fechaInicio posterior al valor fecha del objeto de la clase Evento identificado por idEvento ▪ fechaFin anterior a fechaInicio ▪ fechaFin anterior a la fecha actual
Salida	
Precondiciones	<ul style="list-style-type: none"> ▪ El objeto de la clase Evento identificado por idEvento debe existir
Poscondiciones	<ul style="list-style-type: none"> ▪ Se modificó el atributo periodo_reserva del objeto de la clase Evento identificado por idEvento, siendo las fechas de inicio y fin las indicadas por parámetro

Nombre	cambiarMaximoReservasPersona(idEvento, maximoReservasCliente)
Responsabilidad	Especificar el máximo de reservas permitido por cliente para un evento por parte del encargado
Tipo	GERAT
Notas	
Excepciones	<ul style="list-style-type: none"> ▪ maximoReservasCliente mayor que el valor de plazas_disponibles del objeto de la clase Evento identificado por idEvento
Salida	
Precondiciones	<ul style="list-style-type: none"> ▪ El objeto de la clase Evento identificado por idEvento debe existir
Poscondiciones	<ul style="list-style-type: none"> ▪ Se modificó el atributo máximo_reservas_cliente del objeto de la clase Evento identificado por idEvento, asignándole el valor dado por parámetro

Nombre	idOferta = crearOferta(descuento, tipoOferta)
Responsabilidad	Crear una oferta con un descuento de precio por parte del encargado
Tipo	GERAT
Notas	
Excepciones	<ul style="list-style-type: none"> tipoOferta no tiene un valor válido
Salida	idOferta: identificador del objeto creado de la clase Oferta
Precondiciones	
Poscondiciones	<ul style="list-style-type: none"> Se creó un objeto de la clase oferta debidamente inicializado

Nombre	aplicarOfertaEvento(idEvento, idOferta)
Responsabilidad	Asociar una oferta de días especiales creada previamente a un evento existente
Tipo	GERAT
Notas	
Excepciones	
Salida	
Precondiciones	<ul style="list-style-type: none"> El objeto de la clase Evento identificado por idEvento debe existir El objeto de la clase Oferta identificado por idOferta debe existir
Poscondiciones	<ul style="list-style-type: none"> Se creó un enlace entre el objeto de la clase Evento identificado por idEvento y el objeto de la clase Oferta identificado por idOferta

Nombre	eliminarActividad(idActividad)
Responsabilidad	Eliminar una actividad que no tenga eventos asociados
Tipo	GERAT
Notas	
Excepciones	<ul style="list-style-type: none"> Existen objetos de la clase Evento asociados al objeto de la clase Actividad identificado por idActividad
Salida	
Precondiciones	<ul style="list-style-type: none"> El objeto de la clase Actividad identificado por idActividad debe existir
Poscondiciones	<ul style="list-style-type: none"> Se eliminó el objeto de la clase Actividad identificado por idActividad

Nombre	anularEvento(idEvento)
Responsabilidad	Eliminar un evento que no tenga reservas asociadas
Tipo	GERAT
Notas	
Excepciones	<ul style="list-style-type: none"> Existen objetos de la clase Reserva asociados al objeto de la clase Evento identificado por idEvento
Salida	
Precondiciones	<ul style="list-style-type: none"> El objeto de la clase Evento identificado por idEvento debe existir
Poscondiciones	<ul style="list-style-type: none"> Se eliminó el enlace entre el objeto de la clase Evento identificado por idEvento y el objeto de la clase Actividad asociado a dicho evento Se eliminó el enlace entre el objeto de la clase Evento identificado por idEvento y el objeto de la clase Responsable asociado a dicho evento Para cada objeto de la clase Oferta asociado al objeto de la clase Evento identificado por idEvento: <ul style="list-style-type: none"> Se eliminó el enlace entre el objeto de la clase Oferta y el objeto de la clase Evento Se eliminó el objeto de la clase Oferta Se eliminó el objeto de la clase Evento identificado por idEvento

Nombre	idReembolso, comprobanteReembolso = anularReserva(idReserva, tipo_Pago, cantidad)
Responsabilidad	Anular una reserva reembolsando su pago
Tipo	GERAT
Notas	
Excepciones	<ul style="list-style-type: none"> tipo_Pago es diferente al valor de Tipo_pago del objeto de la clase Pago asociado a la compra de la reserva identificada por idReserva cantidad es distinta al precio del objeto de la clase Reserva identificado por idReserva
Salida	<ul style="list-style-type: none"> idReembolso: identificador del objeto de la clase Reembolso creado comprobanteReembolso: comprobante que se envía al cliente
Precondiciones	<ul style="list-style-type: none"> El objeto de la clase Reserva identificado por idReserva debe existir
Poscondiciones	<ul style="list-style-type: none"> Se eliminó el enlace entre el objeto de la clase Reserva identificado por idReserva y el objeto de la clase Evento asociado a dicha reserva Se eliminó el enlace entre el objeto de la clase Reserva identificado por idReserva y el objeto de la clase Participante asociado a dicha reserva Se eliminó el enlace entre el objeto de la clase Reserva identificado por idReserva y el objeto de la clase Compra asociado a dicha reserva Se eliminó el objeto de la clase Reserva identificado por idReserva Se creó un objeto de la clase Reembolso, unReembolso, debidamente inicializado con la fecha y hora actuales Se creó un enlace entre unReembolso y el objeto de la clase Pago asociado a la compra de la reserva identificada por idReserva