

## PRUEBAS UNITARIAS DAOs

### CartaMenuDAOTest

CP	Nombre del Test
CP1	testFindById_Exists
CP2	testFindById_NotExists
CP3	testSave
CP4	testFindAll
CP5	testDeleteById
CP6	testFindByRestaurante
CP7	testFindByRestaurante_NoMatches

### ItemMenuDAOTest

CP	Nombre del Test
CP1	testFindById_Exists
CP2	testFindById_NotExists
CP3	testSave
CP4	testFindAll
CP5	testDeleteById
CP6	testFindByCartaMenu
CP7	testFindByCartaMenu_NoMatches

### PedidoDAOTest

CP	Nombre del Test
CP1	testFindById_Exists
CP2	testFindById_NotExists
CP3	testSave
CP4	testFindAll
CP5	testDeleteById
CP6	testFindByCliente
CP7	testFindByCliente_NoMatches
CP8	testFindByRestaurante
CP9	testFindByRestaurante_NoMatches

### RestauranteDAOTest

CP	Nombre del Test
CP1	testFindById_Exists
CP2	testFindById_NotExists
CP3	testSave
CP4	testFindAll
CP5	testDeleteById
CP6	testFindByNombreContainingIgnoreCas
CP7	testFindByNombreContainingIgnoreCas

CP8	testFindByDireccion_CalleContainingIgr
CP9	testFindByDireccion_CalleContainingIgr
CP10	testFindByDireccion_MunicipioContaini
CP11	testFindByDireccion_MunicipioContaini
CP12	testFindByDireccion_CodigoPostal
CP13	testFindByDireccion_CodigoPostal_NoM

### UsuarioDAOTest

CP	Nombre del Test
CP1	testFindByld_Exists
CP2	testFindByld_NotExists
CP3	testSave
CP4	testFindAll
CP5	testDeleteByld
CP6	testFindByEmailAndPass_Exists
CP7	testFindByEmailAndPass_NotExists

Condiciones Previas
Mock cartaMenuDAO configurado, CartaMenu con id=1, nombre="Carta 1", restaurante asociado
Mock cartaMenuDAO configurado
Mock cartaMenuDAO configurado, CartaMenu con nombre="Carta 1", restaurante asociado
Mock cartaMenuDAO configurado, Lista con 2 CartaMenu (id=1, id=2)
Mock cartaMenuDAO configurado
Mock cartaMenuDAO configurado, Lista con 2 CartaMenu asociadas a restaurante
Mock cartaMenuDAO configurado, Restaurante sin cartas asociadas

Condiciones Previas
Mock itemMenuDAO configurado, ItemMenu con id=1, nombre="Item 1", precio=10.0, carta asociada
Mock itemMenuDAO configurado
Mock itemMenuDAO configurado, ItemMenu con nombre="Item 1", precio=10.0, carta asociada
Mock itemMenuDAO configurado, Lista con 2 ItemMenu (id=1, id=2)
Mock itemMenuDAO configurado
Mock itemMenuDAO configurado, Lista con 2 ItemMenu asociados a carta
Mock itemMenuDAO configurado, Carta sin items asociados

Condiciones Previas
Mock pedidoDAO configurado, Pedido con id=1, cliente y restaurante asociados
Mock pedidoDAO configurado
Mock pedidoDAO configurado, Pedido con cliente y restaurante asociados
Mock pedidoDAO configurado, Lista con 2 Pedido (id=1, id=2)
Mock pedidoDAO configurado
Mock pedidoDAO configurado, Lista con 2 Pedido asociados a cliente
Mock pedidoDAO configurado, Cliente sin pedidos asociados
Mock pedidoDAO configurado, Lista con 2 Pedido asociados a restaurante
Mock pedidoDAO configurado, Restaurante sin pedidos asociados

Condiciones Previas
Mock restauranteDAO configurado, Restaurante con id=1, nombre="Restaurante A", cif="CIF1", dirección
Mock restauranteDAO configurado
Mock restauranteDAO configurado, Restaurante con nombre="Restaurante Nuevo", cif="CIF3", dirección
Mock restauranteDAO configurado, Lista con 2 Restaurante (id=1, id=2)
Mock restauranteDAO configurado
Mock restauranteDAO configurado, Lista con 2 Restaurante con nombres que contienen "rest"
Mock restauranteDAO configurado, Sin coincidencias para "xyz"

Mock restauranteDAO configurado, Lista con 1 Restaurante con calle que contiene "falsa"
Mock restauranteDAO configurado, Sin coincidencias para "inexistente"
Mock restauranteDAO configurado, Lista con 2 Restaurante con municipios que contienen "madrid"
Mock restauranteDAO configurado, Sin coincidencias para "barcelona"
Mock restauranteDAO configurado, Lista con 1 Restaurante con código postal 28001
Mock restauranteDAO configurado, Sin coincidencias para 99999

Condiciones Previas
Mock usuarioDAO configurado, Usuario (Restaurante) con id=1, email="restaurante@ejemplo.com", pas
Mock usuarioDAO configurado
Mock usuarioDAO configurado, Restaurante con email="nuevo@ejemplo.com", pass="pass789"
Mock usuarioDAO configurado, Lista con 2 Usuario (Restaurante)
Mock usuarioDAO configurado
Mock usuarioDAO configurado, Usuario con email="restaurante@ejemplo.com", pass="pass123"
Mock usuarioDAO configurado, Sin usuario para email="noexistente@ejemplo.com"

Entrada
findById(1L)
findById(1L)
save(CartaMenu)
findAll()
deleteById(1L)
findByRestaurante(restaurante)
findByRestaurante(restaurante2)

Entrada
findById(1L)
findById(1L)
save(ItemMenu)
findAll()
deleteById(1L)
findByCartaMenu(carta)
findByCartaMenu(carta2)

Entrada
findById(1L)
findById(1L)
save(Pedido)
findAll()
deleteById(1L)
findByCliente(cliente)
findByCliente(cliente2)
findByRestaurante(restaurante)
findByRestaurante(restaurante2)

Entrada
findById(1L)
findById(999L)
save(Restaurante)
findAll()
deleteById(1L)
findByNombreContainingIgnoreCase
findByNombreContainingIgnoreCase

findByDireccion_CalleContainingIgnc
findByDireccion_CalleContainingIgnc
findByDireccion_MunicipioContaining
findByDireccion_MunicipioContaining
findByDireccion_CodigoPostal(28001
findByDireccion_CodigoPostal(99999

Entrada
findById(1L)
findById(1L)
save(Restaurante)
findAll()
deleteById(1L)
findByEmailAndPass("restaurante@e
findByEmailAndPass("noexistente@e

Salida Esperada	Resultado
Optional<CartaMenu> con id=1, nombre="Carta 1", restaurante correcto	Éxito
Optional.empty()	Éxito
CartaMenu con id=1, nombre="Carta 1", restaurante correcto	Éxito
Lista con 2 CartaMenu, nombres="Carta 1", "Carta 2"	Éxito
verify(cartasMenuDAO).deleteById(1L) llamado 1 vez	Éxito
Lista con 2 CartaMenu, nombres="Carta 1", "Carta 2", restaurante correcto	Éxito
Lista vacía	Éxito

Salida Esperada	Resultado
Optional<ItemMenu> con id=1, nombre="Item 1", precio=10.0, carta correcta	Éxito
Optional.empty()	Éxito
ItemMenu con id=1, nombre="Item 1", precio=10.0, carta correcta	Éxito
Lista con 2 ItemMenu, nombres="Item 1", "Item 2", precios=10.0, 15.0	Éxito
verify(itemsMenuDAO).deleteById(1L) llamado 1 vez	Éxito
Lista con 2 ItemMenu, nombres="Item 1", "Item 2", precios=10.0, 15.0, carta correcta	Éxito
Lista vacía	Éxito

Salida Esperada	Resultado
Optional<Pedido> con id=1, cliente y restaurante correctos	Éxito
Optional.empty()	Éxito
Pedido con id=1, cliente y restaurante correctos	Éxito
Lista con 2 Pedido, cliente y restaurante correctos	Éxito
verify(pedidosDAO).deleteById(1L) llamado 1 vez	Éxito
Lista con 2 Pedido, cliente correcto	Éxito
Lista vacía	Éxito
Lista con 2 Pedido, restaurante correcto	Éxito
Lista vacía	Éxito

Salida Esperada	Resultado
Optional<Restaurante> con id=1, nombre="Restaurante A", cif="CIF1", calle="Calle A"	Éxito
Optional.empty()	Éxito
Restaurante con id=3, nombre="Restaurante Nuevo", cif="CIF3", calle="Calle Nueva"	Éxito
Lista con 2 Restaurante, nombres="Restaurante A", "Mi Rest"	Éxito
verify(restaurantesDAO).deleteById(1L) llamado 1 vez	Éxito
Lista con 2 Restaurante, nombres="Restaurante A", "Mi Rest"	Éxito
Lista vacía	Éxito

Lista con 1 Restaurante, nombre="Restaurante A", calle="Calle Falsa"	Éxito
Lista vacía	Éxito
Lista con 2 Restaurante, municipios="Madrid", "Madrid Centro"	Éxito
Lista vacía	Éxito
Lista con 1 Restaurante, nombre="Restaurante A", código postal=28001	Éxito
Lista vacía	Éxito

Salida Esperada	Resultado
Optional<Usuario> con id=1, email="restaurante@ejemplo.com", pass="pass123"	Éxito
Optional.empty()	Éxito
Usuario con id=2, email="nuevo@ejemplo.com", pass="pass789"	Éxito
Lista con 2 Usuario, emails="restaurante@ejemplo.com", "restaurante2@ejemplo.com"	Éxito
verify(usuarioDAO).deleteById(1L) llamado 1 vez	Éxito
Optional<Usuario> con email="restaurante@ejemplo.com", pass="pass123"	Éxito
Optional.empty()	Éxito



Condiciones Dominantes
A: Carta existe
A: Carta no existe
A: Carta válida
A: Cartas disponibles
A: ID proporcionado
A: Restaurante con cartas
A: Restaurante sin cartas

Condiciones Dominantes
A: Item existe
A: Item no existe
A: Item válido
A: Items disponibles
A: ID proporcionado
A: Carta con items
A: Carta sin items

Condiciones Dominantes
A: Pedido existe
A: Pedido no existe
A: Pedido válido
A: Pedidos disponibles
A: ID proporcionado
A: Cliente con pedidos
A: Cliente sin pedidos
A: Restaurante con pedidos
A: Restaurante sin pedidos

Condiciones Dominantes
A: Restaurante existe
A: Restaurante no existe
A: Restaurante válido
A: Restaurantes disponibles
A: ID proporcionado
A: Nombre parcial coincide
A: Nombre parcial no coincide

A: Calle parcial coincide
A: Calle parcial no coincide
A: Municipio parcial coincide
A: Municipio parcial no coincide
A: Código postal coincide
A: Código postal no coincide

Condiciones Dominantes
A: Usuario existe
A: Usuario no existe
A: Usuario válido
A: Usuarios disponibles
A: ID proporcionado
A: Credenciales válidas
A: Credenciales inválidas