

Anexo 3: Algoritmos del Back-End

Proyecto: Sistema de Gestión Integral para TECHXPRESS

Este anexo detalla los algoritmos lógicos de los procedimientos PL/SQL principales del back-end: "abrirOrdenTrabajo" (para abrir órdenes de trabajo) y "facturarCliente" (para calcular y actualizar facturas). Estos algoritmos encapsulan la lógica de negocio, incluyendo validaciones, asignaciones automáticas y manejo de excepciones, basados en el código implementado en Oracle Database. Se presentan en pseudo-código indentado para claridad, con estructuras de control (IF, TRY-EXCEPT) y referencias a operaciones de base de datos.

Pseudo-Código Indentado

Procedimiento: abrirOrdenTrabajo

Este procedimiento valida entradas, asigna recursos automáticamente y crea registros relacionados, asegurando atomicidad con COMMIT/ROLLBACK.

PROCEDURE abrirOrdenTrabajo(p_idCliente, p_idEquipo, p_idServicio1, p_idServicio2, p_idServicio3, p_fechaIngreso):

INICIO

DECLARAR variables locales: v_idClienteEquipo, v_idTecnico, v_idFactura, v_idOrden, v_fechaEntrega

DECLARAR excepciones: e_cliente_no_existe, e_equipo_no_existe, e_equipo_no_del_cliente, e_sin_tecnico_disponible

TRY:

// Validar existencia del cliente

CONSULTAR BASE_TABLAS.clientes WHERE idCliente = p_idCliente

SI NO EXISTE:

LANZAR e_cliente_no_existe

// Validar existencia del equipo y pertenencia al cliente

CONSULTAR OPS_TABLAS.equipos WHERE idEquipo = p_idEquipo

OBTENER v_idClienteEquipo

Anexo 3: Algoritmos del Back-End

Proyecto: Sistema de Gestión Integral para TECHXPRESS

SI NO EXISTE:

LANZAR e_equipo_no_existe

SI v_idClienteEquipo != p_idCliente:

LANZAR e_equipo_no_del_cliente

// Calcular fecha de entrega (3 días después del ingreso)

v_fechaEntrega = p_fechaIngreso + 3 DÍAS

// Buscar técnico disponible (sin conflictos de fechas en órdenes existentes)

CONSULTAR BASE_TABLAS.tecnicos t DONDE NO EXISTE (

CONSULTAR OPS_TABLAS.ordenestrabajo ot DONDE ot.idTecnico = t.idTecnico

Y ot.fecha_ingreso <= v_fechaEntrega

Y NVL(ot.fecha_entrega, ot.fecha_ingreso + 3) >= p_fechaIngreso

) Y LIMITAR A 1 RESULTADO

OBTENER v_idTecnico

SI NO ENCONTRADO:

LANZAR e_sin_tecnico_disponible

// Buscar o crear factura para el cliente

CONSULTAR FACT_TABLAS.facturas WHERE clientes_id_cliente = p_idCliente

SI EXISTE:

OBTENER v_idFactura

SINO:

GENERAR v_idFactura = FACT_TABLAS.seq_facturas.NEXTVAL

Anexo 3: Algoritmos del Back-End

Proyecto: Sistema de Gestión Integral para TECHXPRESS

```
INSERTAR EN FACT_TABLAS.facturas (id_factura, fecha_factura, total,  
metodo_pago, clientes_id_cliente)
```

```
VALORES (v_idFactura, p_fechaIngreso, 0, 'PENDIENTE', p_idCliente)
```

```
// Crear orden de trabajo
```

```
GENERAR v_idOrden = OPS_TABLAS.seq_ordenestrabajo.NEXTVAL
```

```
INSERTAR EN OPS_TABLAS.ordenestrabajo (idOrden, idEquipo, idTecnico,  
fecha_ingreso, fecha_entrega, estado, observaciones, idFactura)
```

```
VALORES (v_idOrden, p_idEquipo, v_idTecnico, p_fechaIngreso, v_fechaEntrega,  
'ABIERTA', NULL, v_idFactura)
```

```
// Asociar servicios a la orden (si se proporcionan)
```

```
SI p_idServicio1 NO ES NULL:
```

```
INSERTAR EN OPS_TABLAS.ordenes_servicio (idOrden, idServicio, cantidad)  
VALORES (v_idOrden, p_idServicio1, 1)
```

```
SI p_idServicio2 NO ES NULL:
```

```
INSERTAR EN OPS_TABLAS.ordenes_servicio (idOrden, idServicio, cantidad)  
VALORES (v_idOrden, p_idServicio2, 1)
```

```
SI p_idServicio3 NO ES NULL:
```

```
INSERTAR EN OPS_TABLAS.ordenes_servicio (idOrden, idServicio, cantidad)  
VALORES (v_idOrden, p_idServicio3, 1)
```

```
CONFIRMAR TRANSACCIÓN (COMMIT)
```

```
EXCEPTIONS:
```

```
SI e_cliente_no_existe: LANZAR ERROR -20001 'Orden rechazada: cliente no existe.'
```

```
SI e_equipo_no_existe: LANZAR ERROR -20002 'Orden rechazada: equipo no existe.'
```

Anexo 3: Algoritmos del Back-End

Proyecto: Sistema de Gestión Integral para TECHXPRESS

SI e_equipo_no_del_cliente: LANZAR ERROR -20003 'Orden rechazada: equipo no pertenece al cliente.'

SI e_sin_tecnico_disponible: LANZAR ERROR -20004 'Orden rechazada: no hay técnico disponible.'

PARA OTROS ERRORES: REVERTIR TRANSACCIÓN (ROLLBACK) Y LANZAR ERROR - 20999 CON DETALLES

FIN

Procedimiento: facturarCliente

Este procedimiento calcula el total de una factura basada en servicios asociados y actualiza el registro.

PROCEDURE facturarCliente(p_idCliente):

INICIO

DECLARAR variables locales: v_idFactura, v_totalFactura

DECLARAR excepciones: e_factura_no_existe, e_sin_ordenes

TRY:

// Encontrar factura del cliente

CONSULTAR FACT_TABLAS.facturas WHERE clientes_id_cliente = p_idCliente

OBTENER v_idFactura

SI NO EXISTE:

LANZAR e_factura_no_existe

// Calcular total sumando costos de servicios en órdenes asociadas

CONSULTAR SUMA DE (os.cantidad * s.costos) DESDE OPS_TABLAS.ordenestrabajo ot

UNIENDO OPS_TABLAS.ordenes_servicio os EN ot.idOrden = os.idOrden

Anexo 3: Algoritmos del Back-End

Proyecto: Sistema de Gestión Integral para TECHXPRESS

UNIENDO BASE_TABLAS.servicios s EN os.idServicio = s.idServicio

DONDE ot.idFactura = v_idFactura

OBTENER v_totalFactura

SI v_totalFactura ES NULL O 0:

 LANZAR e_sin_ordenes

// Actualizar factura con total y método de pago

ACTUALIZAR FACT_TABLAS.facturas SET total = v_totalFactura, metodo_pago =
'EFECTIVO'

DONDE id_factura = v_idFactura

CONFIRMAR TRANSACCIÓN (COMMIT)

EXCEPTIONS:

 SI e_factura_no_existe: LANZAR ERROR -20005 'Facturación rechazada: cliente sin
factura.'

 SI e_sin_ordenes: LANZAR ERROR -20006 'Facturación rechazada: no hay órdenes
con servicios.'

 PARA OTROS ERRORES: REVERTIR TRANSACCIÓN (ROLLBACK) Y LANZAR ERROR -
20999 CON DETALLES

FIN

Anexo 3: Algoritmos del Back-End

Proyecto: Sistema de Gestión Integral para TECHXPRESS

