

## MEMORIA

---

### **TERMINAL PUNTO DE VENTA – TPV**

Carlos Castellanos Mateo

Carlos Jiménez Álvarez

Irene Martín Berlanga

Marco Desantes Gutiérrez

Patricia Ortiz Zamora

Víctor Chamizo Rodríguez

# **ÍNDICE**

<b>1. DESCRIPCIÓN DE LA ARQUITECTURA:</b>	<b>1</b>
<b>2. PATRONES</b>	<b>6</b>
<b>3. DISEÑO</b>	<b>7</b>
<b>4. CASOS DE PRUEBA</b>	<b>9</b>

# 1. DESCRIPCIÓN DE LA ARQUITECTURA:

El lenguaje utilizado para dicha aplicación es Java y hacemos uso de SQL para realizar las consultas a la base de datos. Dicha base de datos, es relacional y hemos usado como herramienta MySQL Workbench.

Se ha creado un fichero llamado persistence.xml que sirve para la configuración de las unidades de persistencia.

Si en un futuro se quisiera añadir nuevas entidades, solo bastaría con añadir la clase a dicho documento y ejecutarlo para que posteriormente se cree la tabla en la base de datos asociada.

```
persistence.xml
1 <?xml version="1.0" encoding="UTF-8"?>
2 <persistence version="2.0" xmlns="http://java.sun.com/xml/ns/persistence"
3   <persistence-unit name="ms1718tpvmod">
4     <class>Negocio.Departamento.DepartamentoBO</class>
5     <class>Negocio.Empleado.TemporalBO</class>
6     <class>Negocio.Empleado.FijoBO</class>
7     <class>Negocio.Empleado.EmpleadoBO</class>
8     <class>Negocio.Turno.TurnoBO</class>
9     <class>Negocio.Turno.AsignacionBO</class>
10    <validation-mode>NONE</validation-mode>
11    <properties>
12      <property name="eclipselink.cache.shared.default" value="false"/>
13      <property name="javax.persistence.jdbc.driver"
14        value="com.mysql.jdbc.Driver"/>
15      <property name="javax.persistence.jdbc.url"
16        value="jdbc:mysql://localhost/modelado_de_software"/>
17      <property name="javax.persistence.jdbc.user" value="root"/>
18      <property name="javax.persistence.jdbc.password" value=""/>
19      <!-- enable this property to see SQL and other logging -->
20      <!-- property name="eclipselink.logging.level" value="FINE"/drop-and-crea
21      <property name="eclipselink.ddl-generation" value="none"/>
22      <property name="eclipselink.create-ddl-jdbc-file-name"
23        value="createDDL_ddlGeneration.jdbc"/>
24      <property name="eclipselink.drop-ddl-jdbc-file-name"
25        value="dropDDL_ddlGeneration.jdbc"/>
26      <property name="eclipselink.ddl-generation.output-mode"
27        value="both"/>
28    </properties>
29  </persistence-unit>
30 </persistence>
31
```

Si el nombre de la base de datos o del usuario cambian, se producirían errores de conexión, por lo que no funcionaría la aplicación. Así es, que dichos datos se deben mantener.

El script SQL para la creación de la base de datos se adjuntará en la entrega, guardado con el nombre de ms.sql

Los usuarios para el uso del repositorio son:

<b>ALUMNO</b>	<b>USUARIO</b>	<b>CONTRASEÑA</b>
Carlos Castellanos	ms1718ccastellanos	piscina2540
Carlos Jiménez	ms1718cjimenez	Piscina5477
Irene Martín	ms1718imartin	piscina4575
Marco Desantes	ms1718mdesantes	piscina6257
Patricia Ortiz	ms1718portiz	piscina1574
Victor Chamizo	ms1718vchamizo	piscina3657

Las URLs de los directorios son las siguientes:

- CODIGO  
<https://versiones.fdi.ucm.es/svn/MS/ms1718tpvcod>
- DOCUMENTACIÓN  
<https://versiones.fdi.ucm.es/svn/MS/ms1718tpvdoc>
- MODELADO  
<https://versiones.fdi.ucm.es/svn/MS/ms1718tpvmod>

La aplicación está dividida en paquetes por capas, y a su vez, cada capa dividida en paquetes por módulos:

<b>CAPAS</b>	<b>MÓDULOS</b>		
<i>PRESENTACIÓN</i>	Departamento	Empleado	Turno
<i>NEGOCIO</i>	Departamento	Empleado	Turno

La capa de **presentación** contiene:

- Comandos y Factoría de Comando
- Contexto
- Controlador
- Despachador de Vista
- GUI principal así como la GUI de cada módulo

La capa de **negocio** contiene:

- Servicio de aplicación de cada módulo.
- Objetos de Negocio
- Excepciones de Negocio
- Factoria de Negocio

JPA(Java Persistence API: Gestiona entidades a través del entity manager, que es el persistence manager del JPA.

Los entity Managers se crean y configuran a través de factorias. Y se define a través de la unidad de persistencia.

- Concurrencia
- Objetos de Negocio
- Persistencia

Se ha hecho uso del polimorfismo para poder realizar el cálculo de los suelos de todos los empleados asignados a un departamento, independientemente de que sean fijos o temporales.

## 2. PATRONES

Los patrones utilizados en este proyecto son:

- Modelo-Vista-Controlador (MVC): Patrón que separa los datos de una aplicación, la interfaz de usuario, y la lógica de control en tres componentes distintos.
- Objetos de Negocio: Contiene un conjunto de atributos y valores, operaciones y relaciones con otros objetos del negocio. Los objetos de negocio contienen datos de negocio y conforman el comportamiento del negocio. Cada módulo dispone de una clase de objeto de negocio.
- Controlador de aplicación: Mejora la modularidad del código y la mantenibilidad, facilitando la extensión de la aplicación y la prueba del código de manejo de peticiones de manera independiente del contenedor web.
- Servicio de aplicación: centraliza toda la lógica de negocio. Cada módulo tiene Servicio de Aplicación (SA).
- Factorías abstractas: se encarga de crear las implementaciones correspondientes a cada interfaz.  
Existe una factoría abstracta por capa.
- Singleton: Garantiza que sólo hay una instancia de una clase, proporcionando un único punto de acceso a ella.  
Se usa en el controlador, en las factorías y en la conexión.
- Comandos: Mecanismo que comunica el controlador de aplicación con los servicios de aplicación.
- Despachador: Para que una vista maneje una petición y genere una respuesta, gestionando una cantidad limitada de procesamiento de negocio.  
Se encuentra en la capa de Presentación.
- Contexto: Para evitar utilizar información del sistema específica del protocolo fuera de su contexto relevante.  
Se encuentra en la capa de Integración.

### 3. DISEÑO

Se han realizado los siguientes diagramas UML:

- Diagramas de Clases
- Diagramas de Actividad
- Diagramas de Secuencia
- Diagramas de Caso de Uso

- **Diagramas de Clases**

CAPA	PAQUETE -> CLASE
MAIN	- Main
PRESENTACIÓN	- Comando -> Comando - Contexto - Controlador -> Controlador - Despachador->Despachador - Vistas -> Departamento -> DepartamentoPresentación - Vistas -> Empleado -> EmpleadoPresentacion - Vistas -> Turno -> TurnoPresentacion - Vistas -> VistaPrincipal -> VistaPrincipalPresentacion
NEGOCIO	- Departamento -> DepartamentoNegocio - Empleado -> EmpleadoNegocio - Turno -> TurnoNegocio - Excepciones -> Excepciones - FactoriaNegocio -> FactoriaNegocio

- **Diagramas de Secuencia**

Se han realizado diagramas de secuencia de todas las capas.

**Diagramas de secuencia Departamento**

**CAPA: NEGOCIO**

- CrearDepartamento
- ModificarDepartamento
- BorrarDepartamento
- BuscarDepartamento
- ActivarDepartamento
- CalcularSueldoDepartamento

**CAPA: PRESENTACION**

- CrearDepartamento

**Diagramas de secuencia Empleado**

**CAPA: NEGOCIO**

- CrearEmpleado
- ModificarEmpleado
- BorrarEmpleado
- BuscarEmpleadoPorId
- ActivarEmpleado
- AsignarDepartamento
- ListarEmpleado

**Diagramas de secuencia Turno**

**CAPA: NEGOCIO**

- CrearTurno
- ActivarTurno
- ModificarTurno
- BorrarTurno
- BuscarTurno
- AsignarTurno
- EliminarAsignacion
- ListarTurno
- ModificarTurno



- **Diagramas de Actividad**

Todos los diagramas de actividad han sido revisados y modificados.  
Se realiza un diagrama por cada caso de uso de cada módulo.

- **Diagramas de Casos de Uso**

Diagrama Casos de Uso -> CasoDeUsoDepartamento

Diagrama Casos de Uso -> CasoDeUsoEmpleado

Diagrama Casos de Uso -> CasoDeUsoTurno

#### **4. CASOS DE PRUEBA**

Se ha realizado de manera exhaustiva una lista de casos de prueba en los cuales la aplicación responde positivamente al resultado esperado.

<b><u>DEPARTAMENTO</u></b>		
1	Añadir departamento	Departamento añadido
2	Añadir departamento con mismo nombre	Error
3	Modificar departamento	Departamento modificado
4	Modificar departamento mismo nombre	Error
5	Borrar departamento sin empleados	Departamento borrado
6	Borrar departamento con empleados	Error
7	Calcular Sueldo	Sueldo calculado
8	Calcular Sueldo con Departamento inactivo	Error
9	Buscar Departamento	Departamento encontrado

10	Activar Departamento	Departamento activado
<b>EMPLEADO</b>		
1	Añadir empleado	Empleado añadido
2	Añadir empleado NSS existente	Error
3	Añadir empleado departamento desactivado/inexistente	Error
4	Borrar empleado	Empleado desactivado
5	Modificar empleado NSS existente	Error
6	Modificar empleado departamento desactivado/inexistente	Error
7	Borrar empleado con turno asignado	Error
8	Activar Empleado	Empleado Activado
9	Buscar Empleado	Empleado encontrado
<b>TURNOS</b>		
1	Añadir turno	Turno añadido
2	Añadir turno con hora de entrada y salida existente	Error
3	Modificar turno	Turno Modificado
4	Modificar turno hora de entrada y salida existente	Error
5	Borrar turno con empleado asignado	Error
6	Borrar turno	Turno Borrado
7	Buscar turno	Turno encontrado
<b>ASIGNACIÓN</b>		
1	Añadir asignación	Asignación añadida
2	Añadir asignación empleado inactivo/inexistente	Error

3	Eliminar asignación	Asignación eliminada
4	Añadir asignación turno inactivo/inexistente	Error
5	Añadir asignación turno y empleado inexistente/inactivo	Error