

ANEXO B: MANUAL TECNICO DEL SISTEMA

B.1 Objetivo

El objetivo del manual técnico es dar a conocer los aspectos técnicos del sistema QManagement, de tal manera que los programadores que lean el documento puedan entenderlo con facilidad y puedan realizar actualizaciones o mantenimiento del mismo en caso de que lo requieran.

B.1.1 Diccionario de Datos

B.1.1.1 Nombre de la Tabla: ADVANCE

Descripción: Almacena información de pre registro de clientes.

Columnas	Tipo de Datos	Descripción	Referencia
id	bigint	Identificador del cliente pre registrado	PK
service_id	bigint	Identifica el servicio	FK
advance_time	datetime2(0)	Hola de pre registro	
priority	int	Prioridad del cliente pre registrado	
clients_authorization_id	bigint	Determinar si el cliente está registrado	FK
input_data	nvarchar(150)	Ingresar los datos del cliente	
comments	nvarchar(345)	Ingresar algún comentario por parte del usuario operador	

Tabla B.1.1.1: Descripción de los campos de la tabla Advance

Fuente: Propia

B.1.1.2 Nombre de la Tabla: BREAK

Descripción: Almacena información sobre los recesos de los usuarios.

Columnas	Tipo de Datos	Descripción	Referencia
id	bigint	Identificador del receso	PK
breaks_id	bigint	Registra el id del receso que contiene el nombre	FK
from_time	time(7)	Hora de inicio del receso	
to_time	time(7)	Hora final del receso	

Tabla B.1.1.2: Descripción de los campos de la tabla Break
Fuente: Propia

B.1.1.3 Nombre de la Tabla: BREAKS

Descripción: Almacena la información que describe los descansos de los usuarios.

Columnas	Tipo de Datos	Descripción	Referencia
id	bigint	Identificador del nombre del receso	PK
name	nvarchar(245)	Nombre del receso	

Tabla B.1.1.3: Descripción de los campos de la tabla Breaks
Fuente: Propia

B.1.1.4 Nombre de la Tabla: CALENDAR

Descripción: Almacena la información de los nombres asignados al calendario de trabajo.

Columnas	Tipo de Datos	Descripción	Referencia
id	bigint	Identificador del nombre del calendario	PK
name	nvarchar(45)	Nombre del calendario	

Tabla B.1.1.4: Descripción de los campos de la tabla Calendar
Fuente: Propia

B.1.1.5 Nombre de la Tabla: CALENDAR_OUT_DAYS

Descripción: Almacena la información de las fechas del calendario en las cuales no trabaja el sistema.

Columnas	Tipo de Datos	Descripción	Referencia
id	bigint	Identificador del	PK

		calendario	
out_day	date	Fecha en la que no se trabaja	
calendar_id	bigint	Identificador del nombre del calendario	FK

Tabla B.1.1.5: Descripción de los campos de la tabla Calendar_Out_Days
Fuente: Propia

B.1.1.6 Nombre de la Tabla: CLIENTS

Descripción: Almacena la información detallada de los turnos atendidos por el sistema.

Columnas	Tipo de Datos	Descripción	Referencia
id	bigint	Identificador del registro de atención del cliente	PK
service_id	bigint	Identifica el id del servicio para el que se atendió al cliente	FK
user_id	bigint	Identifica el id del usuario que atendió al cliente	FK
service_prefix	nvarchar(45)	Prefijo o letra del turno atendido	
number	int	Número del turno atendido	
stand_time	datetime2(0)	Tiempo de emisión del turno	
start_time	datetime2(0)	Tiempo de inicio de atención del turno	
finish_time	datetime2(0)	Tiempo de finalización de atención del turno	
clients_authorization_id	bigint	Identifica el id del cliente atendido	FK
result_id	bigint	Identifica la salida generada	FK
input_data	nvarchar(150)	Muestra los datos ingresados por el usuario	
state_in	int	Estado del cliente	

Tabla B.1.1.6: Descripción de los campos de la tabla Clients
Fuente: Propia

B.1.1.7 Nombre de la Tabla: NET

Descripción: Almacena la información sobre los parámetros de conexión del servidor del sistema con los módulos clientes.

Columnas	Tipo de Datos	Descripción	Referencia
id	int	Identificador del registro de información de red	PK
server_port	int	Puerto del servidor	
web_server_port	int	Puerto del servidor web	
client_port	int	Puerto de comunicación del cliente	
finish_time	time	Hora en la que se detiene el servidor	
start_time	time	Hora en la que inicia el funcionamiento del servidor	
version	nvarchar(25)	Versión de la BDD	
first_number	int	Primer número de la secuencia del turno	
last_number	int	Ultimo número de la secuencia del turno	
numering	smallint	Se asigna 0 si todos los servicios van a tener la misma numeración o 1 si cada servicio tiene su propia numeración	
point	int	Se define la descripción del puesto de atención como se va a mostrar	
sound	int	Se define el tipo de configuración para la voz, 0 para no reproducir ningún audio, 1 para mostrar solo las señales y 2 para mostrar señales más la voz	
branch_id	bigint	Identifica el id de la agencia	
sky_server_url	nvarchar(145)	Url del servidor	
zone_board_serv_addr	nvarchar(145)	Dirección del panel del panel del servidor donde se reciben los datos	

zone_board_serv_port	bigint	Puerto del panel del servidor donde se reciben los datos	
voice	int	Reproducir la Voz al mostrar turno	
black_time	int	Tiempo transcurrido de los turnos en la lista negra	
limit_recall	int	Límite de rellamadas	
button_free_design	smallint	Libre diseño de los botones en el punto de entrada	

Tabla B.1.1.7: Descripción de los campos de la tabla Net
Fuente: Propia

B.1.1.8 Nombre de la Tabla: REPORTS

Descripción: Almacena la información de los reportes que emite el sistema.

Columnas	Tipo de Datos	Descripción	Referencia
id	bigint	Identificador del reporte	PK
name	nvarchar(255)	Nombre del reporte	
className	nvarchar(150)	Clase que permite generar el reporte	
template	nvarchar(150)	Plantilla del reporte	
href	nvarchar(150)	Url del reporte	FK

Tabla B.1.1.8: Descripción de los campos de la tabla Reports
Fuente: Propia

B.1.1.9 Nombre de la Tabla: RESPONSE

Descripción: Almacena la información sobre los tipos de respuesta que puede elegir un cliente en la consulta de satisfacción de atención.

Columnas	Tipo de Datos	Descripción	Referencia
id	bigint	Identificador de la respuesta	PK
name	nvarchar(100)	Descripción de la respuesta	
text	nvarchar(500)	Texto de la respuesta	

Tabla B.1.1.9: Descripción de los campos de la tabla Response
Fuente: Propia

B.1.1.10 Nombre de la Tabla: RESPONSE_EVENT

Descripción: Almacena la información receptada en la consulta de satisfacción del cliente.

Columnas	Tipo de Datos	Descripción	Referencia
id	bigint	Identificador de la opinión	PK
resp_date	datetime2(0)	Fecha de la opinión	
response_id	bigint	Identificador del id de la opinión	FK
services_id	bigint	Identificador del servicio que recibió la opinión	FK
users_id	bigint	Identificador del usuario que recibió la opinión	FK
clients_id	bigint	Identificador del id del cliente que emitió la opinión	FK
client_data	nvarchar(245)	Comentario emitido por el cliente	

Tabla B.1.1.10: Descripción de los campos de la tabla Response_Event
Fuente: Propia

B.1.1.11 Nombre de la Tabla: RESULTS

Descripción: Almacena la información sobre el estado de procesamiento de tareas que ejecuta el sistema.

Columnas	Tipo de Datos	Descripción	Referencia
id	bigint	Identificador del resultado	PK
name	nvarchar(150)	Descripción del estado	

Tabla B.1.1.11: Descripción de los campos de la tabla Results
Fuente: Propia

B.1.1.12 Nombre de la Tabla: SCHEDULE

Descripción: Almacena la información sobre el horario de trabajo en el que se encuentra habilitado el sistema.

Columnas	Tipo de Datos	Descripción	Referencia
id	bigint	Identificador del resultado	PK
name	nvarchar(150)	Descripción del estado	
type	int	Tipo de planificación, puede ser semanal o	

		por días	
time_begin_1	time(7)	Hora de inicio	
time_end_1	time(7)	Hora de finalización	
time_begin_2	time(7)	Hora de inicio	
time_end_2	time(7)	Hora de finalización	
time_begin_3	time(7)	Hora de inicio	
time_end_3	time(7)	Hora de finalización	
time_begin_4	time(7)	Hora de inicio	
time_end_4	time(7)	Hora de finalización	
time_begin_5	time(7)	Hora de inicio	
time_end_5	time(7)	Hora de finalización	
time_begin_6	time(7)	Hora de inicio	
time_end_6	time(7)	Hora de finalización	
time_begin_7	time(7)	Hora de inicio	
time_end_7	time(7)	Hora de finalización	
breaks_id1	bigint	Identifica la hora de receso	FK
breaks_id2	bigint	Identifica la hora de receso	FK
breaks_id3	bigint	Identifica la hora de receso	FK
breaks_id4	bigint	Identifica la hora de receso	FK
breaks_id5	bigint	Identifica la hora de receso	FK
breaks_id6	bigint	Identifica la hora de receso	FK
breaks_id7	bigint	Identifica la hora de receso	FK

Tabla B.1.1.12: Descripción de los campos de la tabla Schedule

Fuente: Propia

B.1.1.13 Nombre de la Tabla: SERVICES

Descripción: Almacena la información de los servicios disponibles en la agencia.

Columnas	Tipo de Datos	Descripción	Referencia
id	bigint	Identificador del servicio	PK

name	nvarchar(2000)	Nombre del servicio	
description	nvarchar(2000)	Descripción del servicio	
service_prefix	nvarchar(10)	Prefijo o letra del servicio	
button_text	nvarchar(2500)	El texto que se imprimirá al elegir el servicio	
status	int	Estado del servicio, 1 disponible, 0 no disponible	
enable	int	Si está habilitado o no	
parent_id	bigint	Grupo al que pertenece	
day_limit	int	Restricción de turnos emitidos por día. Si no hay restricción su valor es 0	
person_day_limit	int	Restricción de turnos emitidos por día para un cliente. Si no hay restricción su valor es 0	
advance_limit	int	Limitar el número de clientes pre registrados por hora	
advance_limit_period	int	Limite en días para el pre registro por anticipado. Si no hay restricción su valor es 0	
advance_time_period	int	Periodos en los que se divide un día para el pre registro	
schedule_id	bigint	Identifica el horario de trabajo del servicio	FK
input_required	smallint	Ingreso obligado de información del cliente antes de entrar a la cola	
input_caption	nvarchar(2000)	Ingresar el número de documento al	

		momento de ingresar la información del cliente	
result_required	smallint	El usuario debe ingresar el estado del proceso al atender al cliente	
calendar_id	bigint	Identifica el id del calendario	FK
pre_info_html	nvarchar(MAX)	Texto de información que se muestra en la pantalla	
pre_info_print_text	nvarchar(MAX)	Texto que se imprimirá con el servicio antes del encolamiento	
point	int	Punto de registro del cliente	
ticket_text	nvarchar(1500)	Texto que se imprimirá en el turno	
seq_id	int	Orden de los botones en el punto de registro	
but_x	int	Posición x del botón	
but_y	int	Posición y del botón	
but_b	int	Ancho del botón	
but_h	int	Alto del botón	
deleted	date	Confirmar la fecha de eliminación del servicio	
duration	int	Tiempo promedio para atender el servicio	
sound_template	nvarchar(45)	Plantilla para emitir un sonido al llamar el servicio	
expectation	int	Tiempo de espera obligatorio	

Tabla B.1.1.13: Descripción de los campos de la tabla Services

Fuente: Propia

B.1.1.14 Nombre de la Tabla: SERVICES_USERS

Descripción: Almacena la información de los servicios asignados a los usuarios del sistema.

Columnas	Tipo de Datos	Descripción	Referencia
id	bigint	Identificador del servicio de usuario	PK
services_id	bigint	Identifica el id del servicio	FK
user_id	bigint	Identifica el id del usuario	FK
coefficient	int	Porcentaje de atención	
flexible_coef	smallint	Es posible cambiar el coeficiente de atención	

Tabla B.1.1.14: Descripción de los campos de la tabla Services_Users

Fuente: Propia

B.1.1.15 Nombre de la Tabla: STANDARS

Descripción: Almacena la información de las reglas de negocio establecidas para la atención de los clientes por parte de los asesores de la agencia.

Columnas	Tipo de Datos	Descripción	Referencia
id	bigint	Identificador de la regla de negocio	PK
wait_max	int	Tiempo de espera máximo	
work_max	int	Tiempo de atención máximo	
downtime_max	int	Tiempo máximo de inactividad en la cola	
line_service_max	int	Longitud máxima de la cola para un servicio	
line_total_max	int	El número máximo de clientes para todos los servicios	
relocation	int	Tiempo de traslado de un servicio	

Tabla B.1.1.15: Descripción de los campos de la tabla Standars

Fuente: Propia

B.1.1.16 Nombre de la Tabla: STATISTICS

Descripción: Almacena la información sobre las estadísticas que genera el sistema.

Columnas	Tipo de Datos	Descripción	Referencia
id	bigint	Identificador de la estadística	PK
user_id	bigint	Identifica el id del usuario	FK
client_id	bigint	Identifica el id del cliente	FK
service_id	bigint	Identifica el id del servicio	FK
results_id	bigint	Identifica el estado de la tarea	FK
user_start_time	datetime2(0)	Tiempo de inicio de la atención del usuario con los clientes	
user_finish_time	datetime2(0)	Tiempo de finalización de la atención del usuario con los clientes	
client_stand_time	datetime2(0)	Tiempo en reposo del cliente	
user_work_period	int	Tiempo de trabajo del usuario con el cliente	
client_wait_period	int	Tiempo de espera del cliente	
state_in	int	Estado del cliente	

Tabla B.1.1.16: Descripción de los campos de la tabla Statistics

Fuente: Propia

B.1.1.17 Nombre de la Tabla: USERS

Descripción: Almacena toda la información de los usuarios del sistema.

Columnas	Tipo de Datos	Descripción	Referencia
id	bigint	Identificador del usuario	PK
name	nvarchar(150)	Nombre del usuario	
password	nvarchar(45)	Clave del usuario	

point	nvarchar(45)	Lugar de trabajo	
adress_rs	smallint	Dirección	
enable	int	Estado del usuario	
admin_access	smallint	Si tiene permiso para ingresar al perfil de administrador del sistema	
report_access	smallint	Si tiene permiso para ingresar al perfil de reportería del sistema	
point_ext	nvarchar(1045)	Mostrar puesto en el panel principal de llamadas	
deleted	date	Fecha de confirmación de eliminación	

Tabla B.1.1.17: Descripción de los campos de la tabla Users
Fuente: Propia

B.1.2 Prototipo de Interfaz de Usuario

Una interfaz de usuario debe ser sencilla y de fácil manejo para los usuarios de la aplicación. Debe permitir la comunicación entre el usuario y el computador de forma rápida, intuitiva y transparente.

B.1.2.1 Diseño de la Plantilla Principal

La plantilla principal del Sistema de Gestión del Flujo de Clientes, está estructurada de la siguiente manera:

- **Barra de Títulos:** muestra el nombre del aplicativo y el nombre del usuario logueado en el sistema.
- **Barra de Menú de Opciones:** presenta diferentes opciones que muestran pestañas que permiten ejecutar las diferentes funcionalidades del sistema.
- **Seccion de Contenidos:** abarca la zona principal del sistema, se encuentra ubicada en la parte central de la pantalla.
- **Pie de Página:** contiene un enlace a la página de inicio de la aplicación y

se encuentra ubicada en la parte inferior de la pantalla.

Estado de los Servicios		
SERVICIOS	CLIENTES EN ESPERA	TIEMPO PROMEDIO DE ESPERA
Información	0	00:00:00
Recaudación	0	00:00:00
Cursos Virtuales	0	00:00:00
Cursos Capacitación	0	00:00:00
Cursos Idiomas	0	00:00:00

Figura B.1.2.1: Plantilla Principal
Fuente: Propia

B.1.3 Tecnologías utilizadas en el desarrollo del sistema

Todas las tecnologías utilizadas en el desarrollo del sistema son de libre utilización ya que están regidas bajo los términos de la licencia GPL.

Las tecnologías utilizadas son las siguientes:

- Entorno de Desarrollo Integrado NetBeans.
- Servidor de Aplicaciones Apache Tomcat.
- Plataforma de Desarrollo J2EE.
- Framework Hibernate para la persistencia de datos.
- Framework Spring para la integración de los componentes de la aplicación.
- Framework JavaServer Faces para el diseño de la interfaz de usuario.
- Framework RichFaces para la integración de funcionalidades AJAX en JSF.

- Estándares de código abierto HTML, JavaScript, CSS, Ajax.
- Servidor de Base de Datos SQL Server Express 2008.

B.1.4 Configuración del Proyecto

El proyecto utiliza las tecnologías mencionadas anteriormente y se deben configurar de la siguiente manera:

B.1.4.1 Librerías

- Spring framework 4.0.1 que se debe activar directamente desde las propiedades del proyecto.

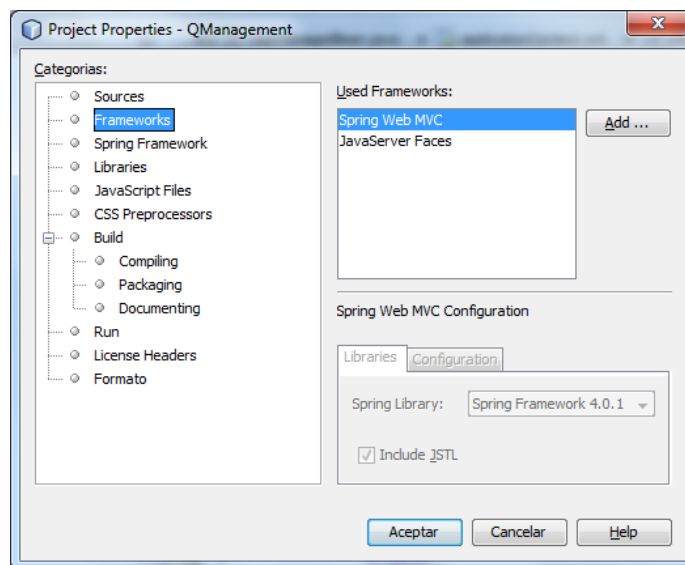


Figura B.1.4.1: Pantalla para agregar el framework Spring
Fuente: Propia

- Framework JavaServer faces 2.2 que se debe activar directamente desde las propiedades del proyecto.

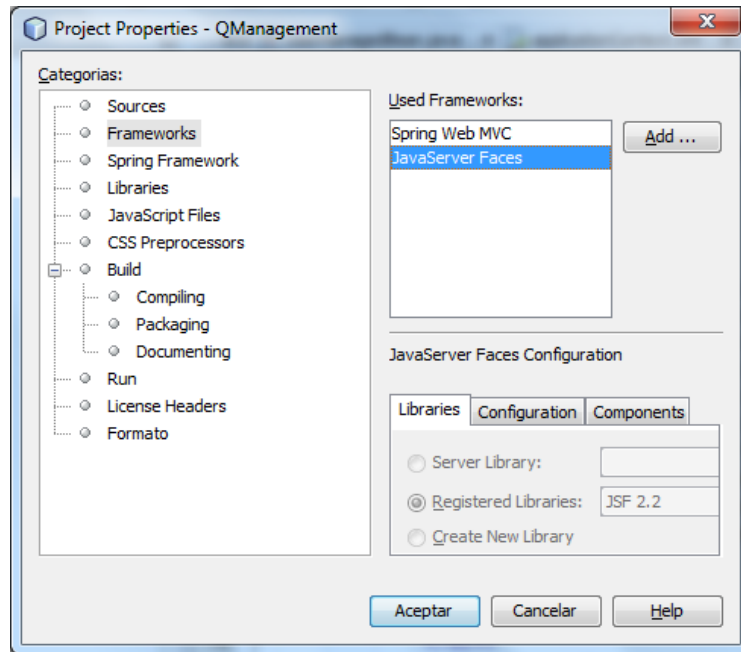


Figura B.1.4.2: Pantalla para agregar el framework JavaServer Faces
Fuente: Propia

- Framework RichFaces 4.2.0, se deben agregar las librerías desde el botón **Add JAR/Folder** y se deben agregar al proyecto.

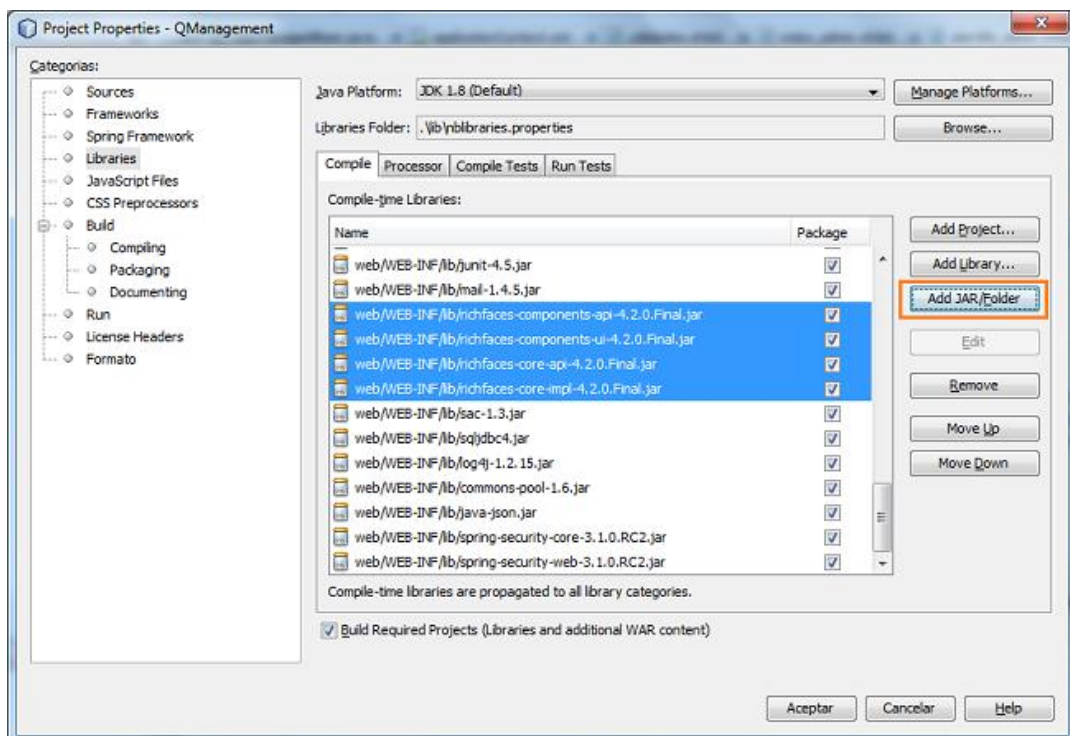


Figura B.1.4.3: Pantalla para agregar el framework RichFaces
Fuente: Propia

- Framework Hibernate 4.3.1, se debe activar directamente desde las propiedades del proyecto.

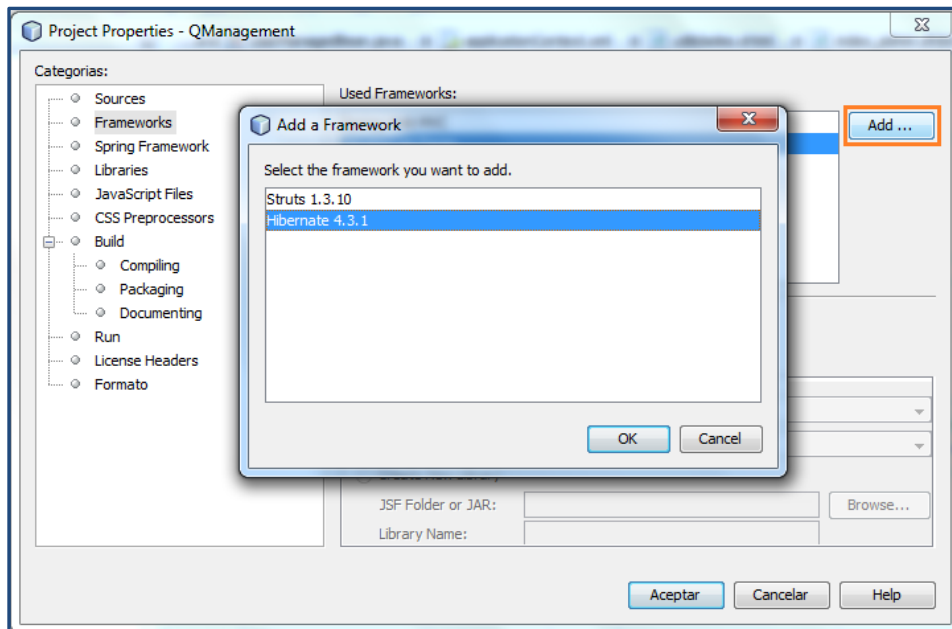


Figura B.1.4.4: Pantalla para agregar el framework Hibernate
Fuente: Propia

B.1.4.2 Archivos de Configuración

- **applicationContext.xml** de Spring en el cual se debe configurar todos los componentes necesarios para la integración de las capas del modelo MVC.

Como se observa en el siguiente fragmento del archivo **applicationContext** QNetDAO se conecta con SessionFactory mediante Spring.

```
<bean id="QNetDAO"  
class="ec.edu.utn.fica.eisic.qmanagement.servidor.dao.QNetDAO">  
  <property name="sessionFactory" ref="SessionFactory" />  
</bean>
```

Para configurar la fábrica de sesiones de hibernate se conecta el bean SessionFactory con la propiedad c3p0DataSource y se anotan las clases para hibernate.


```

<bean id="SessionFactory"
class="org.springframework.orm.hibernate4.LocalSessionFactoryBean"
lazy-init="false" autowire="default">
  <property name="dataSource" ref="c3p0DataSource"/>
  <property name="hibernateProperties">
    <props merge="default">
      <prop
key="hibernate.dialect">org.hibernate.dialect.SQLServerDialect</prop>
      <prop key="hibernate.show_sql">true</prop>
      <prop key="hibernate.format_sql">true</prop>
    </props>
  </property>
  <property name="annotatedClasses">
    <list>
      <value>ec.edu.utn.fica.eisic.qmanagement.servidor.modelo.QNet</value>
    </list>
  </property>
</bean>

```

Se crea el bean c3p0DataSource para asignar los parámetros para el acceso a la fuente de datos.

```

<bean id="c3p0DataSource"
class="com.mchange.v2.c3p0.ComboPooledDataSource" destroy-
method="close">
  <property name="driverClass" value="#{conf.driver}"/>
  <property name="jdbcUrl" value="#{conf.url}"/>
  <property name="user" value="#{conf.user}"/>
  <property name="password" value="#{conf.password}"/>
  <property name="checkoutTimeout" value="10000"/>
  <property name="idleConnectionTestPeriod">
    <value>3600</value>
  </property>
</bean>

```

Se crea el bean conf que permite obtener los valores de los parámetros de conexión a la base de datos desde el archivo configbdd.dat

```

<bean id="conf"
class="ec.edu.utn.fica.eisic.qmanagement.hibernate.AnnotationSessionFact
oryBean"/>

```

Se crea el bean transactionManager para administrar todas las transacciones que realiza hibernate con la base de datos.

```

<bean id="transactionManager"
class="org.springframework.orm.hibernate4.HibernateTransactionManager"
>
  <property name="sessionFactory" ref="SessionFactory" />
</bean>

```

```

<?xml version='1.0' encoding='UTF-8' ?>
<beans xmlns="http://www.springframework.org/schema/beans"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:p="http://www.springframework.org/schema/p"
  xmlns:aop="http://www.springframework.org/schema/aop"
  xmlns:tx="http://www.springframework.org/schema/tx"
  xsi:schemaLocation="http://www.springframework.org/schema/beans http://www.springframework.org/schema/beans/spring-beans-4.0.xsd
  http://www.springframework.org/schema/aop http://www.springframework.org/schema/aop/spring-aop-4.0.xsd
  http://www.springframework.org/schema/tx http://www.springframework.org/schema/tx/spring-tx-4.0.xsd">

  <!-- Declaración de Beans -->
  <bean id="conf" class="ec.edu.utn.fica.eisic.qmanagement.hibernate.AnnotationSessionFactoryBean"/>
  <bean id="QNet" class="ec.edu.utn.fica.eisic.qmanagement.servidor.modelo.QNet"/>

  <!-- Declaración de los beans de servicio -->
  <bean id="QNetService" class="ec.edu.utn.fica.eisic.qmanagement.servidor.servicio.QNetService">
    <property name="qnetDAO" ref="QNetDAO" />
  </bean>
  <!-- Declaración de los beans DAO -->
  <bean id="QNetDAO" class="ec.edu.utn.fica.eisic.qmanagement.servidor.dao.QNetDAO">
    <property name="sessionFactory" ref="SessionFactory" />
  </bean>
  <!-- Declaración de los beans para el acceso a los datos -->
  <bean id="c3p0DataSource" class="com.mchange.v2.c3p0.ComboPooledDataSource" destroy-method="close">
    <property name="driverClass" value="#{conf.driver}"/>
    <property name="jdbcUrl" value="#{conf.url}"/>
    <property name="user" value="#{conf.user}"/>
    <property name="password" value="#{conf.password}"/>
    <property name="checkoutTimeout" value="10000"/>
    <property name="idleConnectionTestPeriod">
      <value>3600</value>
    </property>
  </bean>
  <!-- Fabrica de produccion de sesiones con la BDD. Listado de clases anotadas para hibernate. -->
  <bean id="SessionFactory" class="org.springframework.orm.hibernate4.LocalSessionFactoryBean" lazy-init="false" autowire="default">
    <property name="dataSource" ref="c3p0DataSource"/>
    <property name="hibernateProperties">
      <props merge="default">
        <prop key="hibernate.dialect">org.hibernate.dialect.SQLServerDialect</prop>
        <prop key="hibernate.show_sql">true</prop>
        <prop key="hibernate.format_sql">true</prop>
      </props>
    </property>
    <property name="annotatedClasses">
      <list>
        <value>ec.edu.utn.fica.eisic.qmanagement.servidor.modelo.QNet</value>
        <value>ec.edu.utn.fica.eisic.qmanagement.servidor.modelo.QStandards</value>
        <value>ec.edu.utn.fica.eisic.qmanagement.servidor.modelo.QService</value>
        <value>ec.edu.utn.fica.eisic.qmanagement.servidor.modelo.QServiceLang</value>
        <value>ec.edu.utn.fica.eisic.qmanagement.servidor.modelo.QUser</value>
        <value>ec.edu.utn.fica.eisic.qmanagement.comun.modelo.QCustomer</value>
        <value>ec.edu.utn.fica.eisic.qmanagement.servidor.modelo.QPlanService</value>
        <value>ec.edu.utn.fica.eisic.qmanagement.servidor.modelo.QAdvanceCustomer</value>
        <value>ec.edu.utn.fica.eisic.qmanagement.servidor.modelo.QAuthorizationCustomer</value>
        <value>ec.edu.utn.fica.eisic.qmanagement.servidor.modelo.infosistema.QInfoItem</value>
        <value>ec.edu.utn.fica.eisic.qmanagement.servidor.modelo.respuesta.QRespItem</value>
        <value>ec.edu.utn.fica.eisic.qmanagement.servidor.modelo.respuesta.QRespEvent</value>
        <value>ec.edu.utn.fica.eisic.qmanagement.servidor.modelo.horario.QSchedule</value>
        <value>ec.edu.utn.fica.eisic.qmanagement.servidor.modelo.horario.QBreaks</value>
        <value>ec.edu.utn.fica.eisic.qmanagement.servidor.modelo.horario.QBreak</value>
        <value>ec.edu.utn.fica.eisic.qmanagement.servidor.modelo.resultado.QResult</value>
        <value>ec.edu.utn.fica.eisic.qmanagement.servidor.modelo.calendario.QCalendar</value>
        <value>ec.edu.utn.fica.eisic.qmanagement.servidor.modelo.calendario.FreeDay</value>
      </list>
    </property>
  </bean>
  <!-- Habilitar la configuración del funcionamiento transaccional basado en anotaciones -->
  <tx:annotation-driven transaction-manager="transactionManager" />
  <bean id="transactionManager" class="org.springframework.orm.hibernate4.HibernateTransactionManager">
    <property name="sessionFactory" ref="SessionFactory" />
  </bean>
</beans>

```

Figura B.1.4.5: Contenido del archivo **applicationContext.xml**
Fuente: Propia

- **web.xml** que describe diversas características del archivo WAR.

El elemento `<servlet>` define las características de un Servlet y a su vez está compuesto por los elementos `<servlet-name>` y `<servlet-class>` que indican un nombre corto para el Servlet así como el nombre de la Clase Java que contiene el Servlet respectivamente. En este caso se indica que la Clase llamada `javax.faces.webapp.FacesServlet` será denominada con el nombre `Faces Servlet` que habilita el motor de JSF en el sistema.

```
<servlet>
  <servlet-name>Faces Servlet</servlet-name>
  <servlet-class>javax.faces.webapp.FacesServlet</servlet-class>
  <load-on-startup>1</load-on-startup>
</servlet>
<servlet>
  <servlet-name>dispatcher</servlet-name>
  <servlet-
class>org.springframework.web.servlet.DispatcherServlet</servlet-class>
  <load-on-startup>2</load-on-startup>
</servlet>
```

El elemento `<servlet-mapping>` define la ubicación en términos de directorios de un sitio (URL), esto es, el elemento `<servlet-name>Faces Servlet</servlet-name>` está indicando que el Servlet llamado `Faces Servlet` será accedido cada vez que se accede al directorio base indicado dentro del elemento `<url-pattern>`.

```
<servlet-mapping>
  <servlet-name>Faces Servlet</servlet-name>
  <url-pattern>*.xhtml</url-pattern>
</servlet-mapping>
```

El elemento `<welcome-file-list>` indica que cuando se solicite cualquier directorio sin indicar un archivo en específico se envíe el archivo llamado `index_admin.xhtml`.

```
<welcome-file-list>
  <welcome-file>index_admin.xhtml</welcome-file>
</welcome-file-list>
```

El elemento `<session-config>` indica la duración en minutos de la sesión del usuario cuando navegue en la aplicación.

```
<session-config>
  <session-timeout>
    300
  </session-timeout>
</session-config>
```

El elemento `<listener>` define las características de un evento `ServletContextEvent` que ocurre dentro de la clase `ContextListener`. Definimos el listener `ContextListener` dentro del archivo `web.xml` para que sea la primera clase que se ejecute al momento de levantar la aplicación.

```
<listener>
<listener-
class>ec.edu.utn.fica.eisic.qmanagement.servidor.ContextListener
</listener-class>
</listener>
```

El elemento `<context-param>` permite activar la biblioteca de componentes Richfaces que está dentro del paquete `org.richfaces.skin` donde el valor asignado es el `classic` que corresponde al tipo de presentación de la interfaz de usuario.

```
<context-param>
  <param-name>org.richfaces.skin</param-name>
  <param-value>classic</param-value>
</context-param>
```

Para indicar la ruta del archivo que carga el contexto de la aplicación se debe ingresar en el elemento `<param-value>`.

```
<context-param>
  <param-name>contextConfigLocation</param-name>
  <param-value>/WEB-INF/classes/applicationContext.xml</param-
value>
</context-param>
```

```

<?xml version="1.0" encoding="UTF-8"?>
<web-app version="3.0" xmlns="http://java.sun.com/xml/ns/javaee" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://java.sun.com/xml/ns/javaee http://java.sun.com/xml/ns/javaee/web-app_3_0.xsd">
  <context-param>
    <param-name>org.richfaces.skin</param-name>
    <param-value>classic</param-value>
  </context-param>
  <context-param>
    <param-name>javax.faces.PROJECT_STAGE</param-name>
    <param-value>Development</param-value>
  </context-param>
  <context-param>
    <param-name>contextConfigLocation</param-name>
    <!--<param-value>classpath*:applicationContext.xml</param-value-->
    <param-value>/WEB-INF/classes/applicationContext.xml</param-value>
  </context-param>
  <context-param>
    <param-name>javax.faces.DATETIMECONVERTER_DEFAULT_TIMEZONE_IS_SYSTEM_TIMEZONE</param-name>
    <param-value>true</param-value>
  </context-param>
  <listener>
    <listener-class>org.springframework.web.context.ContextLoaderListener</listener-class>
  </listener>
  <listener>
    <listener-class>ec.edu.utn.fica.eisic.qmanagement.servidor.ContextListener</listener-class>
  </listener>
  <servlet>
    <servlet-name>Faces Servlet</servlet-name>
    <servlet-class>javax.faces.webapp.FacesServlet</servlet-class>
    <load-on-startup>1</load-on-startup>
  </servlet>
  <servlet>
    <servlet-name>dispatcher</servlet-name>
    <servlet-class>org.springframework.web.servlet.DispatcherServlet</servlet-class>
    <load-on-startup>2</load-on-startup>
  </servlet>
  <servlet-mapping>
    <servlet-name>Faces Servlet</servlet-name>
    <url-pattern>*.xhtml</url-pattern>
  </servlet-mapping>
  <servlet-mapping>
    <servlet-name>dispatcher</servlet-name>
    <url-pattern>*.htm</url-pattern>
  </servlet-mapping>
  <session-config>
    <session-timeout>
      300
    </session-timeout>
  </session-config>
  <welcome-file-list>
    <welcome-file>index_admin.xhtml</welcome-file>
  </welcome-file-list>
</web-app>

```

Figura B.1.4.6: Contenido del archivo **web.xml**
Fuente: Propia

- **faces-config.xml** que es el archivo de configuración de JavaServer Faces.

El elemento `<navigation-rule>` permite definir las reglas de navegación de la aplicación.

El elemento `<el-resolver>`

`org.springframework.web.jsf.el.SpringBeanFacesELResolver</el-resolver>`

permite activar el soporte de Spring para la inyección de dependencias de tal forma que JSF sabrá que si no encuentra un bean bajo su contexto debe ir a

buscarlo al contexto de Spring.

```
<application>
  <el-resolver>
    org.springframework.web.jsf.el.SpringBeanFacesELResolver
  </el-resolver>
</application>
```

```
<?xml version='1.0' encoding='UTF-8'?>
<faces-config version="2.1"
  xmlns="http://java.sun.com/xml/ns/javaee"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://java.sun.com/xml/ns/javaee
  http://java.sun.com/xml/ns/javaee/web-facesconfig_2_1.xsd">
  <!-- Integración JSF y Spring -->
  <application>
  <el-resolver>
    org.springframework.web.jsf.el.SpringBeanFacesELResolver
  </el-resolver>
  </application>

  <!-- Configuración de las reglas de navegación -->
  <navigation-rule>
    <from-view-id>/paginas/index.xhtml</from-view-id>
    <navigation-case>
      <from-outcome>success</from-outcome>
      <to-view-id>/paginas/success.xhtml</to-view-id>
    </navigation-case>
    <navigation-case>
      <from-outcome>error</from-outcome>
      <to-view-id>/paginas/error.xhtml</to-view-id>
    </navigation-case>
  </navigation-rule>
</faces-config>
```

Figura B.1.4.7: Contenido del archivo **faces-config.xml**

Fuente: Propia