

**Implementación del proceso de vacaciones de AVANSOFT
bajo una arquitectura orientada a servicios (SOA) utilizando Windows
Workflow Foundation y Biztalk Server 2006**

**David Kepes Hoyos Cód.: 200310077010
Luis Esteban Valencia Cód.: 199710135010**

Trabajo de Grado para optar al título de Ingeniero en Sistemas

**Asesor:
Lenin David Lozano**

**Universidad EAFIT
Departamento de Informática y Sistemas
Medellín
2007**

TABLA DE CONTENIDO

Tabla de contenido.....	2
Tabla de figuras	6
Tabla de TablaS	8
Tabla de codigo	9
Introducción	10
Definición del problema.....	11
Caso de estudio	12
Situación actual.....	12
Propuesta de solución	13
Beneficios.....	14
Mejorar los tiempos de respuesta del proceso de solicitud de vacaciones en general.	14
DIAGRAMA DEL PROCESO DE NEGOCIO	15
3. Especificación de requisitos del Sistema	16
3.1 Requisitos Funcionales.....	16
REQ-FUN-0001 Ingresar solicitud de vacaciones.....	16
REQ-FUN-0002 Aprobar las vacaciones de los empleados.....	17
REQ-FUN-0003 Aprobar la liquidación de las vacaciones.....	17
REQ-FUN-0004 Consultar días festivos.....	17
REQ-FUN-0005 Gestionar días festivos	17
REQ-FUN-0006 Consultar Información de los empleados.....	17
REQ-FUN-0007 Desaprobar las vacaciones de los empleados por Contador.....	18
REQ-FUN-0008 Desaprobar vacaciones de empleados por Jefe Inmediato.....	18
3.2 Requisitos de Información	18
REQ-INF-0001 Información de los días festivos.....	18

REQ-INF-0002 Información de los empleados.....	19
REQ-INF-0003 Información de los periodos de vacaciones disfrutados.....	19
REQ-INF-0004 Información de la solicitud de vacaciones	19
REQ-INF-0005 Información de la notificación de desaprobación.....	20
3.3 Requisitos No Funcionales	20
REQ-NFU-0001 Disponibilidad.....	20
REQ-NFU-0002 Capacidad de cambiar motor de ejecución del flujo de trabajo.....	21
3.4 Requisitos de Negocios	21
REQ-RNG-0001 Calcular días hábiles de las vacaciones	21
REQ-RNG-0002 Número de días antes para solicitud de vacaciones	21
REQ-RNG-0003 Mínimo de días para solicitud	21
5. Modelo de Casos de Uso.....	22
5.1 Modulo administración.....	22
Figura 3. Casos de uso de administración	22
CDU0001 Consultar empleados.....	23
CDU0002 Consultar días festivos.....	23
CDU0003 Ingresar días festivos	24
CDU0004 Actualizar día festivo	24
CDU0005 Eliminar día festivo	25
5.2 Solicitudes de vacaciones	26
Figura 4. Casos de uso de solicitud de vacaciones	27
CDU0006 Eliminar solicitud de vacaciones.....	28
CDU0007 Registrar solicitud de vacaciones	28
CDU0008 Consultar solicitudes de vacaciones registradas	30
CDU0009 Aprobar Vacaciones por Jefe inmediato	30
CDU0010 Desaprobar liquidación de vacaciones	31

CDU0011 Desaprobar vacaciones por Jefe Inmediato	31
CDU0013 Aprobar liquidación de vacaciones.....	32
CDU0014 Consultar solicitudes de vacaciones aprobadas por Jefe inmediato.....	32
modelo de datos	33
descripcion.....	33
diagrama.....	34
DISEÑO	35
Visión general del sistema	35
Objetivos y restricciones.....	35
Requisitos que guían la arquitectura	35
No Funcionales	35
Calidades sistémicas.....	36
Descripción de la solución	38
Decisiones de arquitectura.....	38
Detalle del estilo arquitectónico	39
Vistas de arquitectura.....	39
Vista de Casos de Uso	40
Módulo de administración	40
Solicitud de vacaciones.....	41
Vista de Diseño	42
Engine Factory	42
Descripción.....	42
Diagrama de clases engine factory.....	42
paquetes de diseño arquitecturalmente significativos.....	42
Descripción.....	42
diagrama de paquetes	44

Vista de procesos	44
diagrama de secuencia – creación del motor.....	45
Descripción.....	45
Diagrama	45
Vista de componentes	46
Descripción.....	46
diagrama de componentes.....	46
Vista de despliegue	47
Descripción de los nodos.....	47
Diagrama de despliegue	48
Interfaces.....	48
WsNomina	49
WsVacaciones.....	50
WsNotificaciones	52
Plataforma tecnológica.....	55
IMPLEMENTACION	55
Implementación en Windows WorkFlow Foundation	60
Implementación en BiztalkServer	70
manual de instalacion	82
TOMA DE DECISION.....	96
Tabla de toma de decisiones.....	97
Conclusiones de la toma de decisión	98
Conclusiones	100
Bibliografía	102

TABLA DE FIGURAS

Figura 1. Diagrama de proceso de negocio	15
Figura 2. Especificación de los requisitos del sistema	16
Figura 3. Casos de uso de administración	22
Figura 4. Casos de uso de solicitud de vacaciones	27
Figura 5. Diagrama de modelo de datos	34
Figura 6. Casos de uso módulo administración	40
Figura 7. Casos de uso modulo solicitud vacaciones	41
Figura 8. Diagrama de clases Engine Factory	42
Figura 9. Diagrama de paquetes.....	44
Figura 10. Diagrama de secuencia de instanciación del motor de procesos	46
Figura 11. Diagrama de componentes.....	47
Figura 12. Diagrama de despliegue	48
Figura 13. Diagrama de infraestructura de red y servidores	56
Figura 14. Menú del analista.....	57
Figura 15. Menú del director	57
Figura 16. Menú del contador	57
Figura 17. Diagrama de flujo de trabajo Workflow Foundation 1	62
Figura 18. Diagrama de flujo de trabajo Workflow Foundation 2	63
Figura 19. Esquema de archivos de solicitud vacaciones.....	71
Figura 20. Diagrama del flujo de trabajo de ingresar solicitud en Biztalk.....	74
Figura 21. Esquema de archivos de cambio solicitud jefe	74
Figura 22. Flujo de trabajo de cambio de solicitud contador jefe en Biztalk.....	78
Figura 23. Esquema de archivos de cambio solicitud contador	78

Figura 24. Diagrama del flujo de trabajo de cambio de solicitud contador	82
Figura 25. Instalación paso 1	83
Figura 26. Instalación paso 2	84
Figura 27. Instalación paso 3	84
Figura 28. Instalación paso 4	85
Figura 29. Instalación paso 5	86
Figura 30. Instalación paso 6	86
Figura 31. Instalación paso 7	87
Figura 32. Instalación paso 8	87
Figura 33. Instalación paso 9	88
Figura 34. Instalación paso 10	89
Figura 35. Instalación paso 11	89
Figura 36. Instalación paso 12	90
Figura 37. Instalación paso 13	91
Figura 38. Instalación paso 14	91
Figura 39. Instalación paso 15	92
Figura 40. Instalación paso 16	93
Figura 41. Instalación paso 17	94
Figura 42. Instalación paso 18	95
Figura 43. Desinstalación	96

TABLA DE TABLAS

Tabla 1. Requisitos que guían la arquitectura.....	36
Tabla 2. Calidades sistémicas	38
Tabla 3. Contrato del servicio WsNomina.....	50
Tabla 4. Contrato del servicio WsVacaciones.....	52
Tabla 5. Contrato del servicio WsNotificaciones	54
Tabla 6. Actividades de WorkFlow Foundation.....	61
Tabla 7. Tabla de toma de decisiones de criterios.....	98

TABLA DE CODIGO

Código 1. Interfaz IFactory.....	58
Código 2. Iniciar WorkFlow	58
Código 3. Aprobar vacaciones jefe	59
Código 4. Desaprobar vacaciones jefe	59
Código 5. Aprobar vacaciones contador	60
Código 6. Desaprobar vacaciones contador	60
Código 7. Implementaion de IniciarWorkFlow.....	65
Código 8. Application Start en el Global.asax.....	66
Código 9. Interfaz IWFVacaciones para eventos del workflow	66
Código 10. Implementación de AprobarVacacionesJefe	67
Código 11. Evento AprobarSolicitudJefe	68
Código 12. FolderEntrada en web.config.....	70
Código 13. Ejemplo del esquema de archivos de solicitud vacaciones	71
Código 14. FolderCambioEstadoJefe en el web.config	74
Código 15. Ejemplo del esquema de archivos para cmbio de solicitud jefe.....	75
Código 16. Extracción de variables	75
Código 17. FolderCambioEstadoContador en el web.config	78
Código 18. Ejemplo del esquema de archivos de cambio solicitud contador.....	79
Código 19. Extracción de variables del esquema de cambio de solicitud contador	79
Código 20. Crear base de datos	82
Código 21. Usar base de datos.....	82

INTRODUCCIÓN

Hoy en día las empresas manejan complejos portafolios de sistemas de información, que se han ido construyendo con el tiempo en diferentes plataformas, utilizando variadas tecnologías, paradigmas de programación y hardware diverso, éstos son islas aisladas y cubren un proceso específico del negocio.

Una arquitectura orientada a servicios (SOA) propone una solución a este problema, utilizando servicios que se comuniquen entre ellos mismos siguiendo una interfaz definida y estos a su vez podrían luego conectarse para formar aplicaciones que cubran todo el proceso de negocio de la compañía, a su vez los servicios expuestos por cada aplicación podrían ser utilizados por otros sistemas sin incurrir en el error de tener varios sistemas que arrojan la misma información y que realizan la misma funcionalidad.

“Por muchos años, SOA ha prometido entregar eficiencia y costos de Infraestructura tecnológica, definiendo una metodología para usar y reusar componentes de software y procesos de negocio. Sin embargo, SOA es aún nuevo y las organizaciones están todavía en el proceso de aprender cómo se debe implementar de tal manera que utilicen todo su potencial tanto para servicios al interior como al exterior de la empresa”. [1]

DEFINICIÓN DEL PROBLEMA

El proceso de las vacaciones en Avansoft involucra diversas áreas y procesos de la compañía, muchos de éstos son ejecutados manualmente por diferentes personas que tienen que interactuar con varios sistemas y esto genera lentitud y falta de seguimiento detallado de cada paso del flujo de trabajo.

Por otra parte los involucrados nunca reciben ningún tipo de notificación del estado del proceso ni de las actividades que fueron, van a ser o están pendientes por realizar.

Aunque Avansoft cuenta con diferentes aplicativos que apoyan ciertos procesos de la organización estos se encuentran aislados y no existe ningún tipo de interoperabilidad entre ellos, lo cual hace que puedan producirse errores humanos y disminuyen la productividad de los empleados.

CASO DE ESTUDIO

SITUACIÓN ACTUAL

Avansoft es una empresa de desarrollo de software a la medida con 10 años en el mercado y clientes como la mediana y grande empresa.

Actualmente el proceso de vacaciones de Avansoft es realizado mediante un flujo de trabajo manual por los empleados de las diferentes áreas involucradas, como analistas, directores, gestión humana y contador.

Este proceso se ha tornado demasiado lento e ineficiente para las personas que dependen del tiempo de respuesta para poder programar su tiempo de descanso o vacaciones, pueden pasar días sin que se reciba ningún tipo de información, notificación o respuesta de la solicitud de vacaciones de un empleado, lo cual ha hecho que se pierda la confianza y la credibilidad en el proceso por parte de las personas implicadas.

El proceso de solicitud de las vacaciones actualmente se realiza de la siguiente manera:

1. El empleado debe descargar un formato de la Intranet para la solicitud de vacaciones, este lo debe diligenciar con las fechas deseadas de período de vacaciones.
2. El empleado debe entregar el formato a su debido jefe inmediato.
3. El jefe inmediato de acuerdo a su cronograma del proyecto actual decide si aprueba o rechaza la solicitud para los días indicados.
4. En caso tal de que la solicitud sea aprobada por el jefe inmediato este debe hacerla llegar al área de gestión humana.
5. El área de gestión humana debe verificar que la solicitud cumpla con los requisitos exigidos, los cuales son:
 - a. La solicitud debe tener los datos correctamente diligenciados.

- b. La solicitud no puede ser entregada con menos de 5 días antes del período de disfrute de las vacaciones.
 - c. El empleado debe tener los días disponibles para poder disfrutar de las vacaciones.
6. El área de gestión humana debe entregar la solicitud aprobada al contador.
 7. El contador procede a liquidar las vacaciones y a generar la carta de entrega de las vacaciones.

PROPUESTA DE SOLUCIÓN

Se pretende realizar una solución basada en la tecnología Microsoft para suprimir el uso de formatos en papel, debido a que Microsoft posee dos soluciones para implementar procesos de negocio como este: Windows Workflow Foundation y Biztalk Server. Se plantearán los criterios necesarios para posteriormente realizar la comparación de las tecnologías que permitan implementar flujos de trabajo.

La solución tendrá una interfaz web para:

- Ingreso de solicitud de vacaciones.
- Aprobación o rechazo de las solicitudes de vacaciones por parte del jefe inmediato.
- Aprobación o rechazo de la liquidación de las solicitudes aprobadas por el jefe inmediato por parte del contador.
- Maestro de días festivos.
- Consultar el estado de las solicitudes.

Se podrá realizar monitoreo de las solicitudes en sus diferentes estados y se enviarán notificaciones por correo electrónico y por Spark (el mensajero interno de la compañía).

BENEFICIOS

Mejorar los tiempos de respuesta del proceso de solicitud de vacaciones en general.

Cuando la solución esté completamente implementada los tiempos de respuesta del proceso de solicitud de vacaciones serán mejores debido a que se ahorrará tiempo por parte de los empleados al no tener que distribuir un papel a lo largo de la organización.

Reducir costos de papelería

Con la solución se pretende eliminar por completo el uso del papel y los costos de impresión de los formatos.

Aumentar la confianza de los empleados

La solución aumentará la confianza de los empleados brindándoles un sistema que les permita monitorear sus solicitudes de vacaciones, adicional a esto las solicitudes no se traspapelan con otros documentos e incrementa la productividad de las personas implicadas en el proceso.

DIAGRAMA DEL PROCESO DE NEGOCIO

Este modelo permite capturar el esquema general y procesos que gobiernan el negocio, proporciona una visión de donde el sistema propuesto podría apoyar el negocio y las actividades diarias, también justifica la construcción del sistema capturando los procesos manuales y automatizados actuales que serán subidos al nuevo sistema, y ayuda a identificar los beneficios y costos asociados.

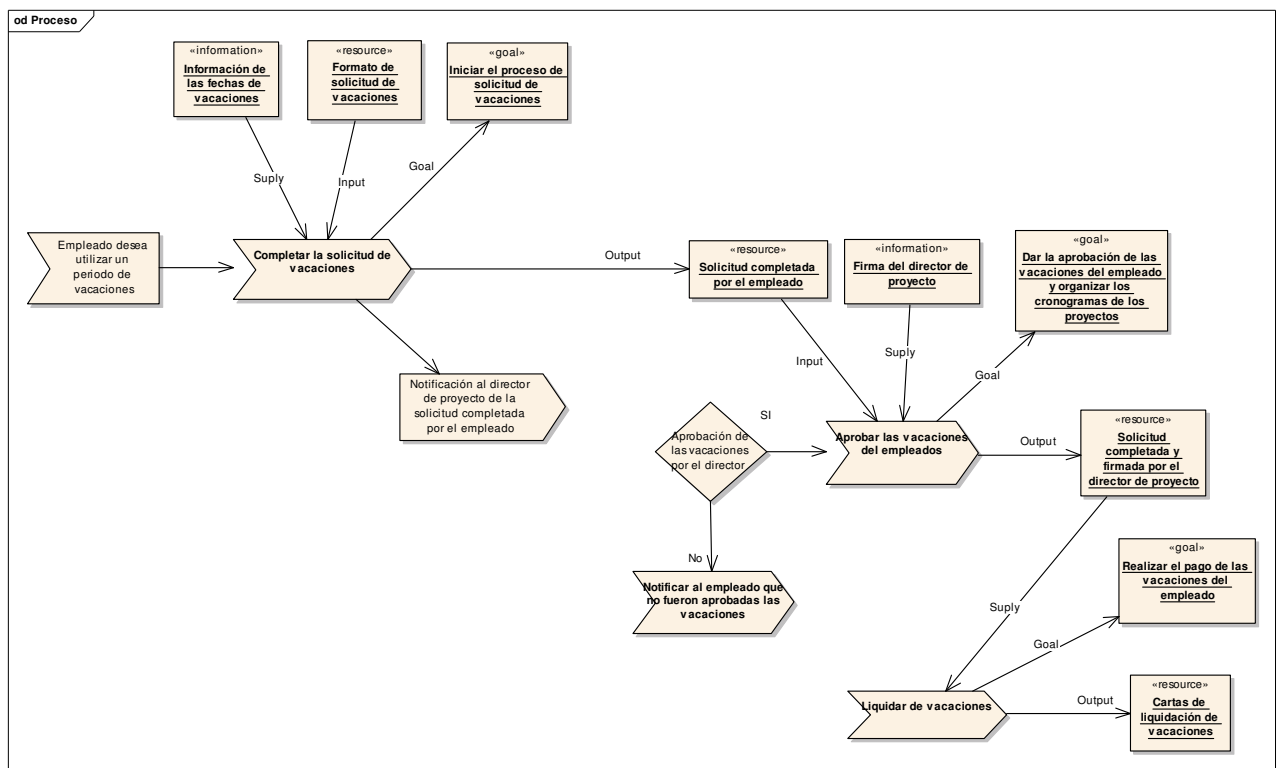


Figura 1. Diagrama de proceso de negocio

3. ESPECIFICACIÓN DE REQUISITOS DEL SISTEMA

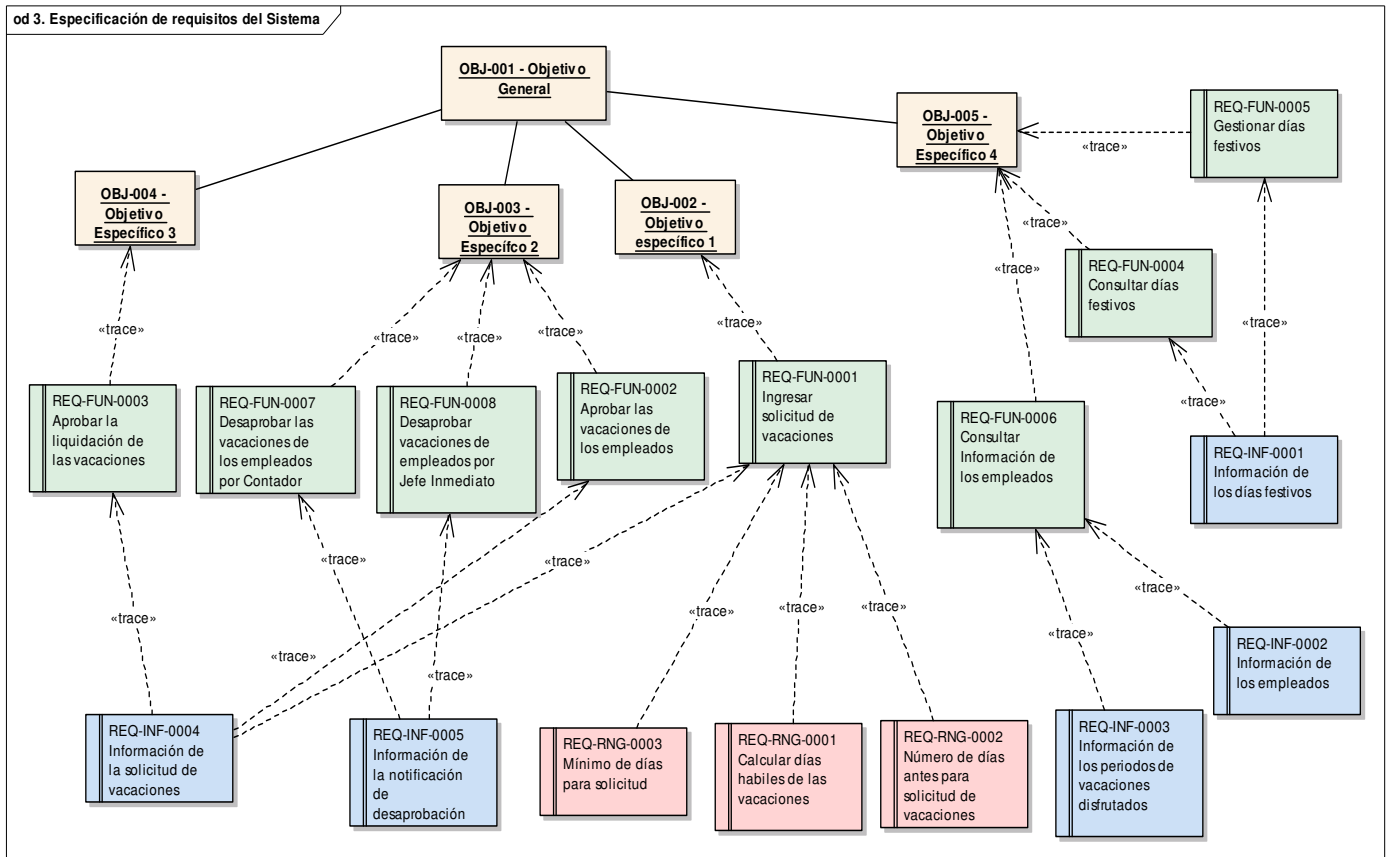


Figura 2. Especificación de los requisitos del sistema

3.1 REQUISITOS FUNCIONALES

REQ-FUN-0001 INGRESAR SOLICITUD DE VACACIONES

El sistema deberá permitir al empleado realizar la solicitud de sus vacaciones y enviar una notificación vía email al jefe inmediato.

Fuente: David Arango

Fecha: Julio 15 de 2007

REQ-FUN-0002 APROBAR LAS VACACIONES DE LOS EMPLEADOS

El sistema deberá permitir al jefe inmediato aprobar las vacaciones del empleado que realizó la solicitud y enviar notificación vía email al empleado y al contador.

Fuente: David Arango
Fecha: Julio 15 de 2007

REQ-FUN-0003 APROBAR LA LIQUIDACIÓN DE LAS VACACIONES

El sistema deberá permitir al contador aprobar la liquidación de las vacaciones de los empleados aprobadas previamente por el jefe inmediato

Fuente: David Arango
Fecha: Julio 15 de 2007

REQ-FUN-0004 CONSULTAR DÍAS FESTIVOS

El sistema deberá permitir consultar los días festivos de un servicio web.

Fuente: David Kepes
Fecha: Julio 15 de 2007

REQ-FUN-0005 GESTIONAR DÍAS FESTIVOS

El sistema deberá permitir al administrador ingresar, actualizar y eliminar los días festivos del año.

Fuente: David Arango
Fecha: Julio 15 de 2007

REQ-FUN-0006 CONSULTAR INFORMACIÓN DE LOS EMPLEADOS

El sistema deberá permitir al administrador consultar la información de los empleados.

Fuente: David Arango
Fecha: Julio 15 de 2007

REQ-FUN-0007 DESAPROBAR LAS VACACIONES DE LOS EMPLEADOS POR CONTADOR

El sistema deberá permitir al Contador desaprobado las vacaciones de los empleados.

Se enviará una notificación al empleado y jefe inmediato con la decisión tomada y su correspondiente causa.

Fuente: David Arango
Fecha: Julio 15 de 2007

REQ-FUN-0008 DESAPROBAR VACACIONES DE EMPLEADOS POR JEFE INMEDIATO

El sistema deberá permitir al jefe inmediato desaprobado las vacaciones de los empleados a cargo.

Se enviará una notificación al empleado con la decisión tomada y su correspondiente causa.

Fuente: David Arango
Fecha: Julio 15 de 2007

3.2 REQUISITOS DE INFORMACIÓN

REQ-INF-0001 INFORMACIÓN DE LOS DÍAS FESTIVOS

El sistema debe guardar los días festivos por cada año.

- Fecha
- Nombre del día festivo

Fuente: David Arango
Fecha: Julio 15 de 2007

REQ-INF-0002 INFORMACIÓN DE LOS EMPLEADOS

El sistema debe guardar la información de los empleados.

- Cédula
- Nombres y apellidos
- Email
- Sexo
- Fecha de ingreso
- Jefe Inmediato
- Usuario

Fuente: David Arango
Fecha: Julio 15 de 2007

REQ-INF-0003 INFORMACIÓN DE LOS PERIODOS DE VACACIONES DISFRUTADOS

El sistema debe guardar la información de las vacaciones de los empleados:

- Cédula empleado
- Días disfrutados
- Fecha Inicial
- Fecha Final

Fuente: David Arango
Fecha: Julio 15 de 2007

REQ-INF-0004 INFORMACIÓN DE LA SOLICITUD DE VACACIONES

El sistema debe guardar la información de las solicitudes de las vacaciones:

- Cédula empleado
- Fecha solicitud
- Fecha Inicial vacaciones
- Fecha Final
- Días Hábiles Solicitados (Campo calculado)
- Estado (Ingresada, Aprobada por jefe, Aprobada por Contador, Desaprobada por jefe, Desaprobada por Contador)
- PagoVacaciones

Fuente: David Arango

Fecha: Julio 15 de 2007

REQ-INF-0005 INFORMACIÓN DE LA NOTIFICACIÓN DE DESAPROBACIÓN

El sistema debe guardar la información de las solicitudes de las vacaciones:

- Cédula empleado
- Fecha solicitud
- Fecha Inicial vacaciones
- Fecha Final
- Días Hábiles Solicitados (Campo calculado)
- Estado (Ingresada, Aprobada por jefe, Aprobada por Contador, Desaprobada por jefe, Desaprobada por Contador)
- PagoVacaciones

Fuente: David Arango

Fecha: Julio 15 de 2007

3.3 REQUISITOS NO FUNCIONALES

REQ-NFU-0001 DISPONIBILIDAD

El sistema debe estar como mínimo disponible en el horario laboral de Avansoft.
7:00-6:00PM

Fuente: David Arango
Fecha: Julio 15 de 2007

REQ-NFU-0002 CAPACIDAD DE CAMBIAR MOTOR DE EJECUCIÓN DEL FLUJO DE TRABAJO

El sistema debe ser parametrizable, de tal forma que se pueda elegir si se requiere utilizar Windows Workflow Foundation o BizTalk Server para la ejecución de los procesos.

Fuente: David Kepes
Fecha: Julio 15 de 2007

3.4 REQUISITOS DE NEGOCIOS

REQ-RNG-0001 CALCULAR DÍAS HÁBILES DE LAS VACACIONES

Se deben calcular los días hábiles en pantalla cuando el empleado este realizando la solicitud de vacaciones.

Fuente: David Arango
Fecha: Julio 15 de 2007

REQ-RNG-0002 NÚMERO DE DÍAS ANTES PARA SOLICITUD DE VACACIONES

El sistema no deberá permitir ingresar solicitudes de vacaciones si la diferencia entre la fecha de solicitud y la fecha de inicio de las vacaciones es de menos de 15 días.

Fuente: David Arango
Fecha: Julio 15 de 2007

REQ-RNG-0003 MÍNIMO DE DÍAS PARA SOLICITUD

El sistema sólo debe permitir solicitudes de vacaciones mayores a 5 días.

Fuente: David Arango
Fecha: Julio 15 de 2007

5. MODELO DE CASOS DE USO

5.1 MODULO ADMINISTRACIÓN

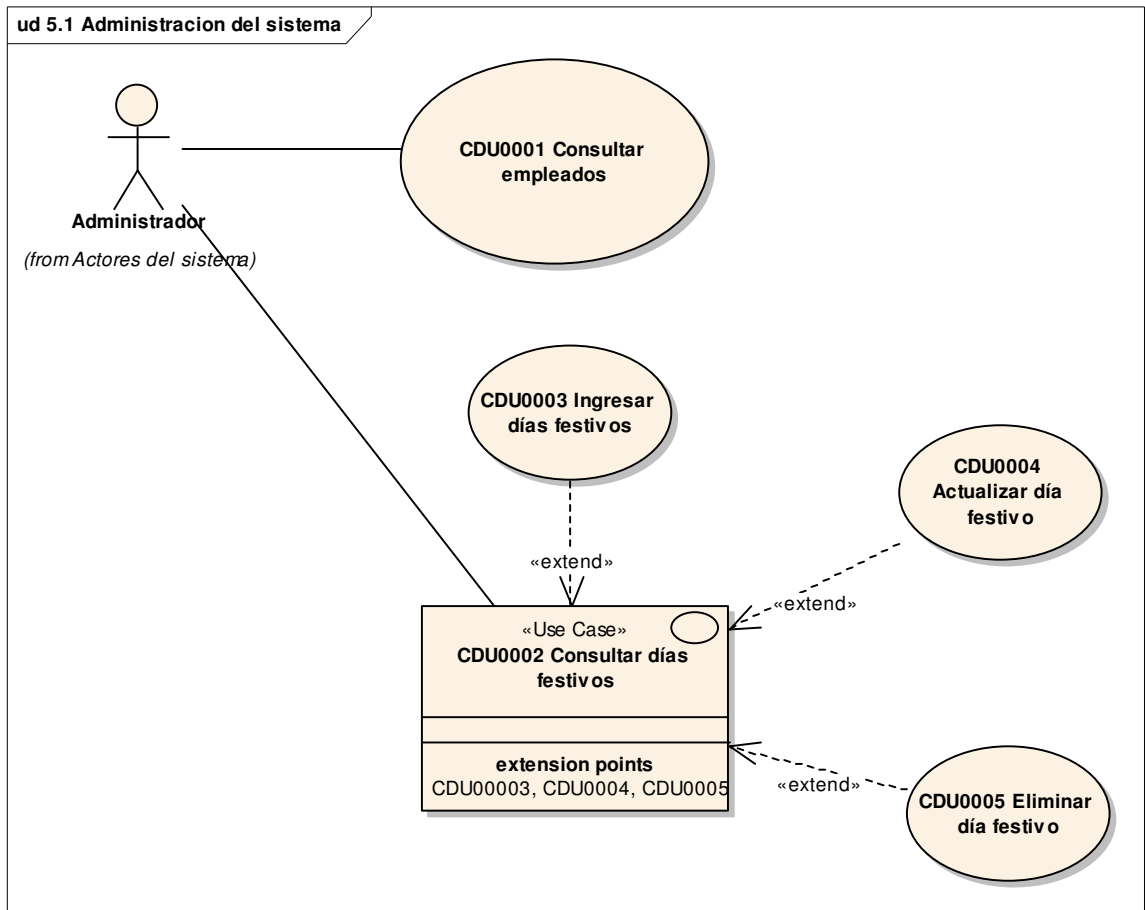


FIGURA 3. CASOS DE USO DE ADMINISTRACIÓN

CDU0001 CONSULTAR EMPLEADOS

Permite ver los empleados registrados en el sistema

Condiciones

- *Aprobado Pre-condición.* El usuario debe haber seleccionado la opción Empleados.
- *Aprobado Post-condición.* El sistema despliega la consulta de empleados.

Escenarios

Flujo básico {Básico}.

1. El sistema despliega el listado de empleados ordenados por apellidos de manera descendente.

- Apellidos
- Nombres
- Cédula

CDU0002 CONSULTAR DÍAS FESTIVOS

Permite consultar los días festivos

Condiciones

- *Aprobado Pre-condición.* El usuario debe haber seleccionado la opción Administrar días festivos.
- *Aprobado Post-condición.* El sistema despliega la consulta de días festivos.

Escenarios

Flujo básico {Básico}.

1. El sistema despliega los días festivos:

- Fecha
- Descripción día festivo.

2. El sistema despliega la opción de ingresar, actualizar o eliminar

CDU0003 INGRESAR DÍAS FESTIVOS

Permite ingresar días festivos

Condiciones

- *Aprobado Pre-condición.* El usuario debe haber seleccionado la opción ingresar día festivo.
- *Aprobado Post-condición.* El sistema ingresa la información del día festivo.

Escenarios

Flujo básico {Básico}.

1. El administrador del sistema elige ingresar día festivo.
2. El sistema despliega el formulario de ingreso:
 - Fecha
 - Descripción
3. El administrador del sistema ingresa la información y elige guardar.
4. El sistema guarda la información y despliega el mensaje "Día festivo ingresado satisfactoriamente"

3.1 La fecha ya existe en la base de datos {Alternativo}.

3.1.1 El sistema despliega el mensaje "La fecha digitada ya aparece registrada en la base de datos."

3.1.2 El caso de uso retorna al punto 2 del flujo básico

CDU0004 ACTUALIZAR DÍA FESTIVO

Permite actualizar los días festivos

Condiciones

- *Aprobado Pre-condición.* El usuario debe haber seleccionado la opción de actualizar día festivo.
- *Aprobado Post-condición.* El sistema actualiza la información del día festivo.

Escenarios

Flujo básico {Básico}.

1. El Administrador del sistema elige un día festivo y selecciona actualizar.
2. El sistema despliega el formulario de actualización:
 - Fecha
 - Descripción
3. El administrador del sistema digita la información y selecciona guardar.
4. El sistema guarda el día festivo actualizado y despliega el mensaje "Día festivo actualizado satisfactoriamente".

3.1 La fecha ya existe en la base de datos {Alternativo}.

3.1.1 El sistema despliega el mensaje "La fecha digitada ya aparece registrada en la base de datos."

3.1.2 El caso de uso retorna al punto 2 del flujo básico.

CDU0005 ELIMINAR DÍA FESTIVO

Permite eliminar un día festivo

Condiciones

- *Aprobado Pre-condición.* El usuario debe haber seleccionado la opción eliminar día festivo.

- *Aprobado Post-condición.* El sistema elimina el día festivo.

Escenarios

Flujo básico {Básico}.

1. El administrador del sistema elige un registro y selecciona eliminar.
2. El sistema despliega el mensaje "Esta seguro que desea eliminar el día festivo"
3. El administrador del sistema elige SI.
4. El sistema elimina el día festivo y despliega el mensaje "Día festivo eliminado satisfactoriamente"

3.1 El administrador del sistema elije la opción No {Alternativo}.

- 3.1.1 El sistema cancela la operación

5.2 SOLICITUDES DE VACACIONES

ud 5.2 Solicitudes de vacaciones

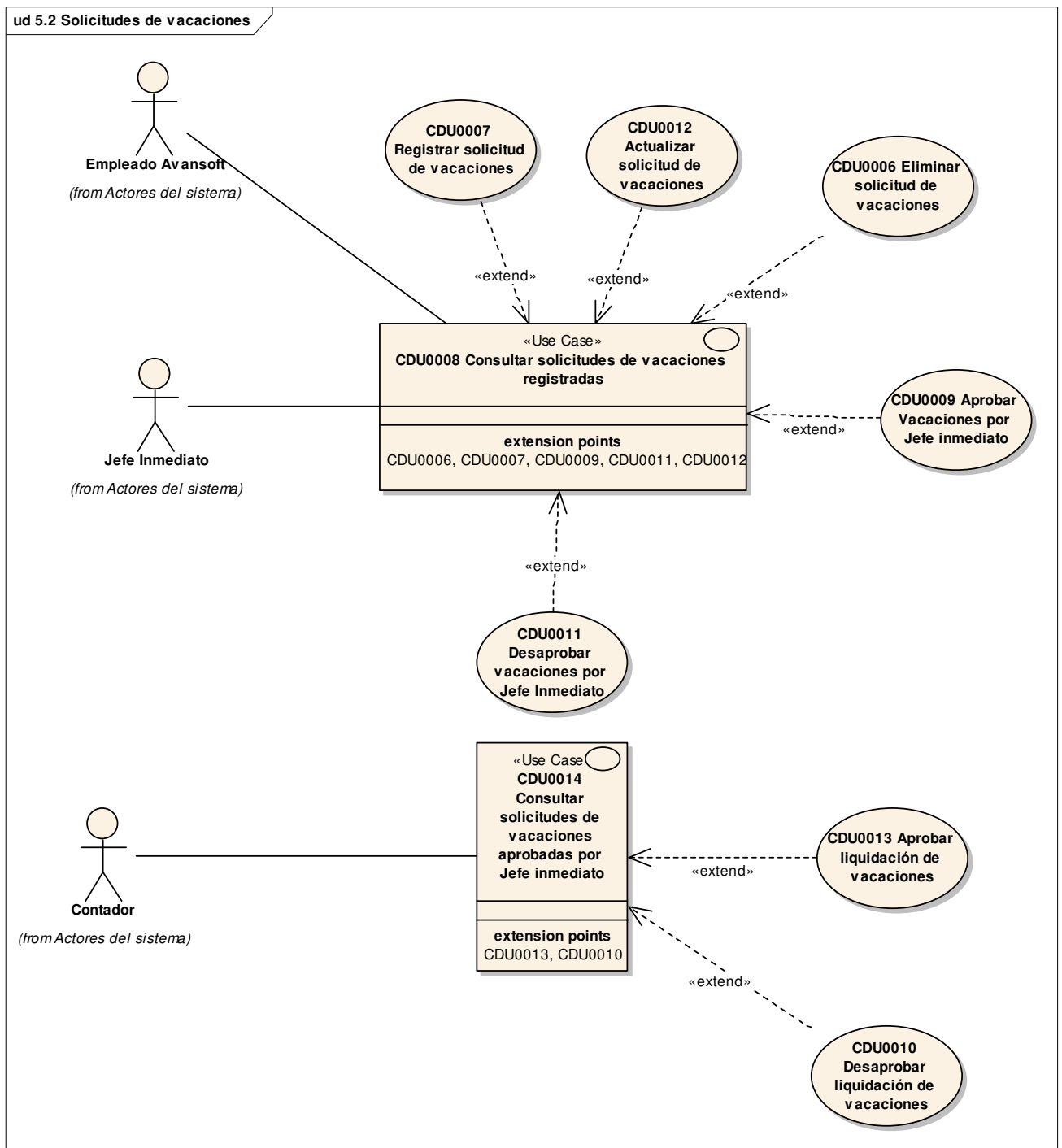


FIGURA 4. CASOS DE USO DE SOLICITUD DE VACACIONES

CDU0006 ELIMINAR SOLICITUD DE VACACIONES

Permite eliminar una solicitud de vacaciones.

Condiciones

- *Aprobado Pre-condición.* El usuario debe haber seleccionado la opción registrar solicitud de vacaciones.
- *Aprobado Post-condición.* El sistema guarda la información de la solicitud de vacaciones.

Escenarios

Flujo básico {Básico}.

1. El usuario selecciona la solicitud y elige eliminar.
2. El sistema despliega el mensaje "Esta seguro que desea eliminar la solicitud?"
3. El usuario elige SI.
4. El sistema elimina la solicitud de la base de datos y despliega el mensaje "Solicitud eliminada satisfactoriamente".

CDU0007 REGISTRAR SOLICITUD DE VACACIONES

Permite registrar una solicitud de vacaciones.

Condiciones

- *Aprobado Pre-condición.* El usuario debe haber seleccionado la opción registrar solicitud de vacaciones.
- *Aprobado Post-condición.* El sistema guarda la información de la solicitud de vacaciones.

Escenarios

Flujo básico {Básico}.

1. El usuario selecciona la opción registrar-
2. El sistema despliega el formulario de solicitud de vacaciones.
 - Fecha Inicial
 - Fecha Final
 - Pago de vacaciones (inmediato, nómina)
 - Días (se calcula automáticamente restando Fecha final - Fecha inicial)
3. El usuario selecciona las fechas y elige guardar.
4. El sistema guarda en la base de datos la siguiente información.
 - Id del empleado
 - Fecha inicial
 - Fecha final
 - Días
 - Estado (almacena por defecto el estado "ingresado")
 - Pago vacaciones
 - Fecha de solicitud (fecha actual)
5. El sistema envía un correo electrónico al jefe inmediato con el siguiente texto. "Ha sido ingresada una nueva solicitud".

3.1 La fecha de inicio no sobrepasa 15 días con respecto a la fecha de solicitud {Alternativo}.

3.1.1 El sistema muestra el mensaje "La fecha de inicio de vacaciones debe ser mínimo 15 días después de la fecha actual".

3.1.2. El sistema retorna al punto 2 del flujo básico.

3.2. El número de días es menor o igual a 5 {Alternativo}.

3.2.1. El sistema muestra el mensaje "El número de días mínimo que usted puede disfrutar de vacaciones es 6".

3.2.2. El sistema retorna al punto 2 del flujo básico.

CDU0008 CONSULTAR SOLICITUDES DE VACACIONES REGISTRADAS

Permite consultar las solicitudes de vacaciones

Condiciones

- *Aprobado Pre-condición.* El usuario debe seleccionar la opción de consultar vacaciones registradas.
- *Aprobado Post-condición.* El sistema despliega las vacaciones registradas al Jefe inmediato.

Escenarios

Flujo básico {Básico}.

1. El sistema despliega las solicitudes de vacaciones con los siguientes campos:
 - Cédula
 - Apellidos
 - Nombres
 - Fecha Inicial Vacaciones
 - Fecha Final Vacaciones
2. El sistema despliega las opciones Registrar, Actualizar y Eliminar.

CDU0009 APROBAR VACACIONES POR JEFE INMEDIATO

Permite aprobar las vacaciones por parte del Jefe inmediato

Condiciones

- *Aprobado Pre-condición.* El usuario debe haber seleccionado la opción aprobar vacaciones.
- *Aprobado Post-condición.* El sistema actualiza la información del estado de la solicitud.

Escenarios

Flujo básico {Básico}.

1. El Jefe inmediato selecciona un registro y elige aprobar.

2. El estado cambia a "Aprobada por Jefe".
3. El sistema envía un correo electrónico al solicitante con el texto "La solicitud ha sido aprobada por el jefe inmediato".
4. El sistema envía un correo electrónico al contador con el siguiente texto: " La solicitud ha sido aprobada por el jefe inmediato".

CDU0010 DESAPROBAR LIQUIDACIÓN DE VACACIONES

Permite desaprobado las vacaciones de un empleado por parte del Contador

Condiciones

- *Aprobado Pre-condición.* El usuario debe haber seleccionado la opción desaprobado vacaciones.
- *Aprobado Post-condición.* El sistema guarda la información del estado de la solicitud.

Escenarios

Flujo básico {Básico}.

1. El Contador elige una solicitud y selecciona Desaprobado.
2. El sistema cambia el estado a Desaprobado por Contador.
3. El sistema envía un correo electrónico al empleado y el jefe inmediato con el siguiente texto; "La solicitud ha sido desaprobada por el jefe inmediato".

CDU0011 DESAPROBAR VACACIONES POR JEFE INMEDIATO

Permite desaprobado las vacaciones de un empleado

Condiciones

- *Aprobado Pre-condición.* El usuario debe haber selecciona la opción desaprobado vacaciones.
- *Aprobado Post-condición.* El sistema guarda la información del estado de la solicitud actualizada.

Escenarios

Flujo básico {Básico}.

1. El Jefe Inmediato elige una solicitud y selecciona Desaprobar.
2. EL sistema cambia el estado a "Desaprobada por jefe".
3. El sistema envía un correo electrónico al empleado con el siguiente texto:
"Su solicitud de ha sido desaprobada".

CDU0013 APROBAR LIQUIDACIÓN DE VACACIONES

Permite aprobar la liquidación de las vacaciones aprobadas por los jefes inmediatos.

Condiciones

- *Aprobado Pre-condición.* El usuario debe haber seleccionado la opción Aprobar liquidación de vacaciones.
- *Aprobado Post-condición.* El Sistema actualiza la información del estado de la solicitud de las vacaciones.

Escenarios

Flujo Básico {Básico}.

1. El Contador selecciona un registro y elige aprobar.
2. El estado cambia a Aprobada por Contador.
3. El sistema envía un correo electrónico al solicitante y Jefe inmediato con el texto " La solicitud ha sido aprobada por el contador".

CDU0014 CONSULTAR SOLICITUDES DE VACACIONES APROBADAS POR JEFE INMEDIATO

Permite consultar las solicitudes aprobadas para procesar su liquidación

Condiciones

- *Aprobado Pre-condición.* El usuario debe haber seleccionado la opción

- de consultar vacaciones aprobadas.
- *Aprobado Post-condición.* El sistema despliega las solicitudes de vacaciones aprobadas por los jefes inmediatos.

Escenarios

Flujo básico {Básico}.

1. El sistema despliega las vacaciones aprobadas con los siguientes datos:
 - Nombre solicitante
 - Fecha de ingreso
 - Fecha solicitud de inicio de vacaciones
 - Fecha solicitud de fin de vacaciones
 - Días hábiles totales
 - Pago de vacaciones
2. El sistema despliega las opciones: Aprobar liquidación de vacaciones y Desaprobar liquidación de vacaciones

MODELO DE DATOS

DESCRIPCION

Este modelo de datos refleja cómo se persistirá la información del proceso de vacaciones de Avansoft.

tblEmpleados: Contiene la información básica de los empleados.

tblPeriodosDisfrutados: en esta tabla se guarda la información de fechas de los períodos disfrutados de cada empleado para poder realizar los cálculos de días sin disfrutar.

tblSolicitudVacaciones: contiene la información básica de las solicitudes.

tblCambiosSolicitud: contiene la información de los cambios de estado de las solicitudes.

tblDiaFestivo: contiene la lista de los días festivos por año.

DISEÑO

La siguiente sección del presente documento es basada en el formato original de Avansoft F_DOC_28 Arquitectura de software el cual contiene las secciones necesarias para explicar la arquitectura de la solución propuesta.

VISIÓN GENERAL DEL SISTEMA

Se requiere construir un sistema de información para apoyar el proceso de vacaciones de AVANSOFT.

OBJETIVOS Y RESTRICCIONES

- Parametrizar la lista de días festivos por año.
- Presentar a los usuarios implicados en el proceso información oportuna mediante interfaces gráficas del sistema y mediante envío de notificaciones a correo y Spark.
- Parametrizar el flujo de trabajo si se desea utilizar con Windows Workflow Foundation o con Biztalk Server.

REQUISITOS QUE GUÍAN LA ARQUITECTURA

No Funcionales

Nro. Requisito:	Descripción
REQ-NFU-0094	El sistema debe ser parametrizable, de tal forma que se puede elegir si se requiere utilizar Windows Workflow Foundation o Biztalk Server para la

Nro. Requisito:	Descripción
	ejecución de los procesos.

Tabla 1. Requisitos que guían la arquitectura

CALIDADES SISTÉMICAS.

Atributo de Calidad	Descripción	Métrica
Desempeño	Expectativas de usuario en cuanto a desempeño.	El aplicativo en su proceso más importante que es el ingresar una solicitud no se debe demorar más de 3 segundos. El tiempo máximo de carga de las páginas o formularios debe ser de máximo 4 segundos.
Confiabilidad	Probabilidad de Falla de un componente. Estabilidad de los datos. La IEEE la define como "Capacidad del sistema o componente a desempeñar sus funciones bajo condiciones establecidas en un período de tiempo dado.	Se espera que el sistema tenga como máximo por mes un error de confiabilidad en la información, es decir una solicitud mal ingresada o con información corrupta.
Disponibilidad	Porcentaje de tiempo que el sistema está disponible	El aplicativo debe estar disponible de lunes a

Atributo de Calidad	Descripción	Métrica
	para su uso. Se ve afectada por factores más allá de la arquitectura. Un corte de energía, una bomba, etc.	viernes en horario laboral de Avansoft, es decir de 7:30AM a 530PM
Escalabilidad	Habilidad de adicionar capacidad y más usuarios en el tiempo. Usualmente la escalabilidad implica un mayor número de recursos.	El sistema debe permitir un número máximo de 10 usuarios concurrentes con un crecimiento del 10% anual durante 3 años.
Seguridad	Incluye niveles de autenticación y autorización soportados, granularidad de controles de autorización, mecanismos para proveer auditoria, técnicas usadas para asegurar la integridad de recursos, y resistencia al acceso no autorizado.	El aplicativo debe implementar seguridad por roles, es decir el empleado solo puede ingresar solicitudes, el jefe solo puede aprobar o rechazarlas y el contador solo puede aprobar o rechazar para la liquidación de estas.
Flexibilidad	Cuán fácil es el sistema a proponer nuevos o diferentes servicios	Se espera que ante cambios de ley o políticas internas de la compañía que requieran modificación del aplicativo se impacte máxima en un

Atributo de Calidad	Descripción	Métrica
		día adicional a la complejidad de los nuevos servicios que se deban implementar.
Reusabilidad	Habilidad de usar componentes o servicios individuales en la construcción de aplicaciones adicionales.	El sistema debe permitir la exposición de los servicios para que estos puedan ser consumidos por otras aplicaciones.
Usabilidad	Hacer la interfaz fácil de usar.	El tiempo de un usuario para aprender a dominar una opción es de máximo una hora.

Tabla 2. Calidades sistémicas

DESCRIPCIÓN DE LA SOLUCIÓN

Decisiones de arquitectura

Para satisfacer el requerimiento de escalabilidad, se usará una arquitectura por capas que en un futuro podría dividirse por servidores para mejorar el rendimiento con muchos usuarios.

Para poder satisfacer el requerimiento no funcional REQ-NFU-0094 Capacidad de cambiar motor de ejecución del flujo de trabajo, se implementará el patrón Factory, el cual permitirá en tiempo de ejecución cambiar el motor de ejecución de procesos sin afectar la solución del sistema.

Detalle del estilo arquitectónico

El sistema será diseñado para una arquitectura de 6 capas.

Características

- Facilita la descomposición del problema en varios niveles de abstracción.
- Soporta fácilmente la evolución del sistema; los cambios sólo afectan a las capas vecinas
- Se pueden cambiar las implementaciones respetando las interfaces con las capas adyacentes.
- Se pueden usar servicios creados anteriormente para otros proyectos sin necesidad de volverlos a codificar.

VISTAS DE ARQUITECTURA

“El modelamiento, implementación y documentación de un sistema requiere que este sea visto desde diferentes perspectivas. Por este motivo, la arquitectura del sistema será representada en una aproximación de 4 + 1 vistas, originalmente planteado por Kruchten, pero modificadas de la siguiente manera:

- *Vista de Casos de Uso: Su principal propósito es definir cuáles son las directrices principales del sistema, determinadas por los requisitos del sistema.*
- *Vista de Diseño: Esta vista contiene cualquier definición del sistema así como los diagramas de clases o entidad relación que representen los servicios que el sistema proveerá. Aquí se describe las partes arquitectónicamente significativas del modelo de diseño, tal como su descomposición en subsistemas y paquetes.*
- *Vista de Procesos: Esta vista desplegará los procesos que forman los mecanismos del sistema. Estos serán representados como colaboraciones, secuencias y diagramas de actividades.*

- *Vista de Componentes: Esta vista incluirá las especificaciones de interfaz de usuario y de sistema, es decir, los diferentes componentes que conforman el sistema.*
- *Vista de Despliegue: Esta vista representará como los diferentes nodos de hardware del sistema que convivirán y como cada uno de estos nodos estará instalado y desplegado.”*

VISTA DE CASOS DE USO

Módulo de administración

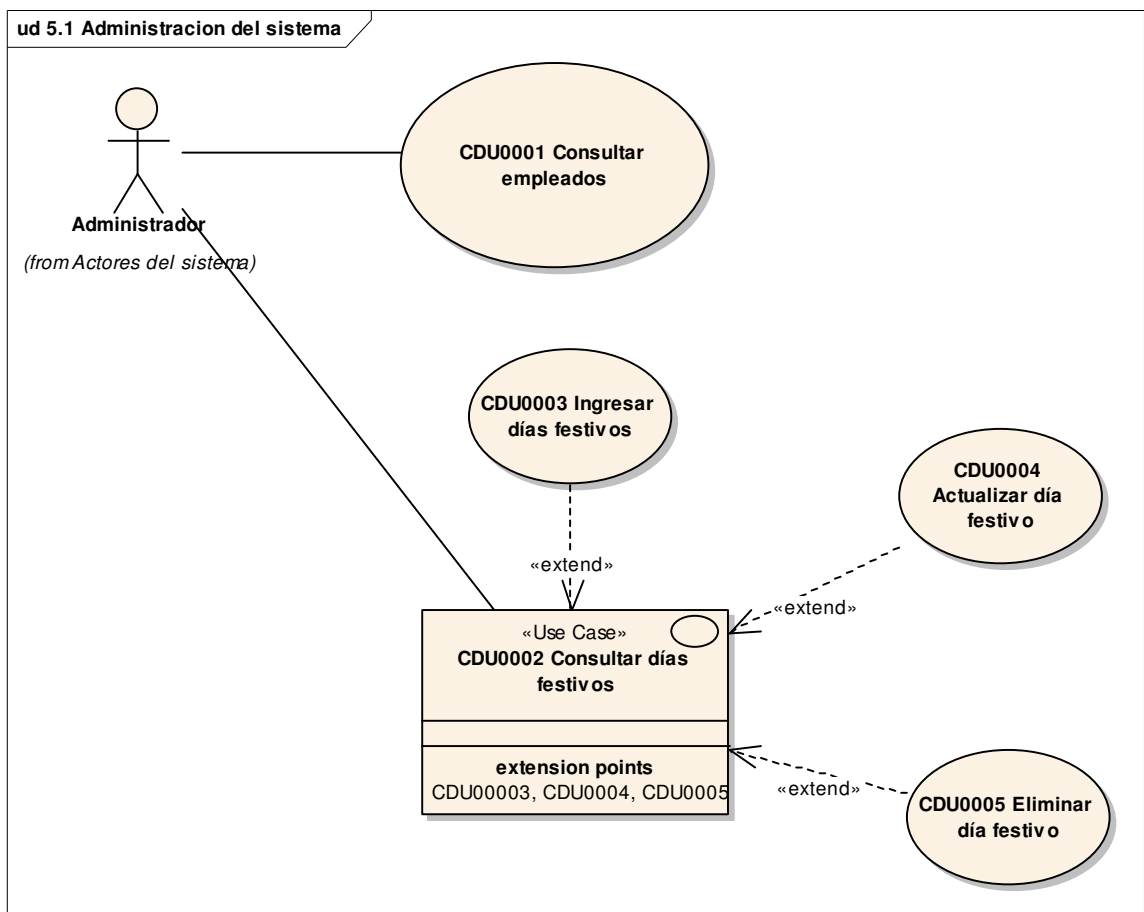


Figura 6. Casos de uso módulo administración

Solicitud de vacaciones

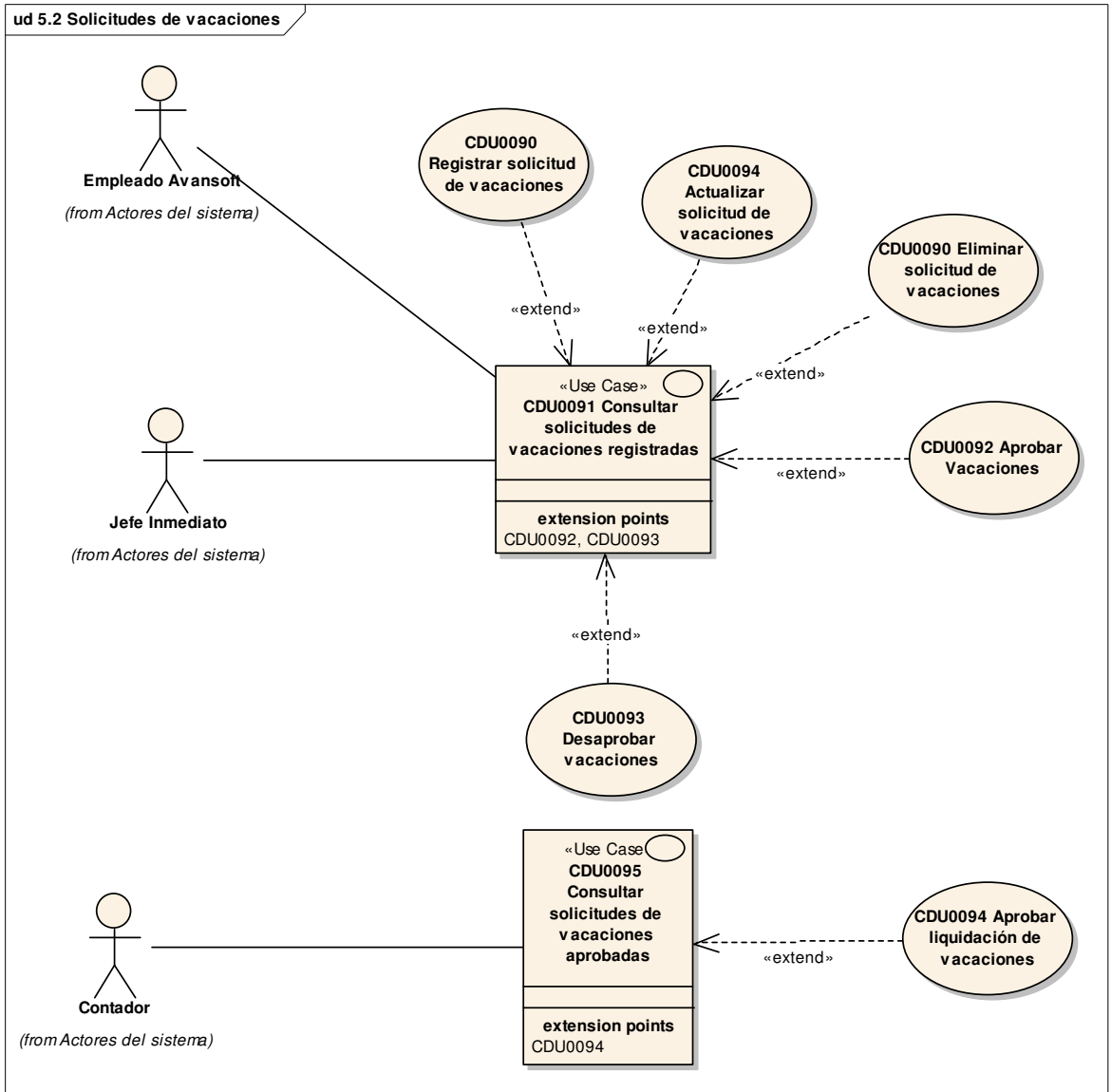


Figura 7. Casos de uso modulo solicitud vacaciones

VISTA DE DISEÑO

Engine Factory

Descripción

Para poder implementar el requisito no funcional REQ-NFU-0094 Capacidad de cambiar motor de ejecución del flujo de trabajo, se utilizará un Factory, el cual llamamos **EngineFactory** esta clase se encargará de consultar en el archivo de configuración de la aplicación cuál motor es el que se desea usar para iniciar el proceso de solicitud de vacaciones, una vez consultado, creará una clase concreta ya sea de **EngineBiztalk** o **EngineWF** según este configurado.

Diagrama de clases engine factory

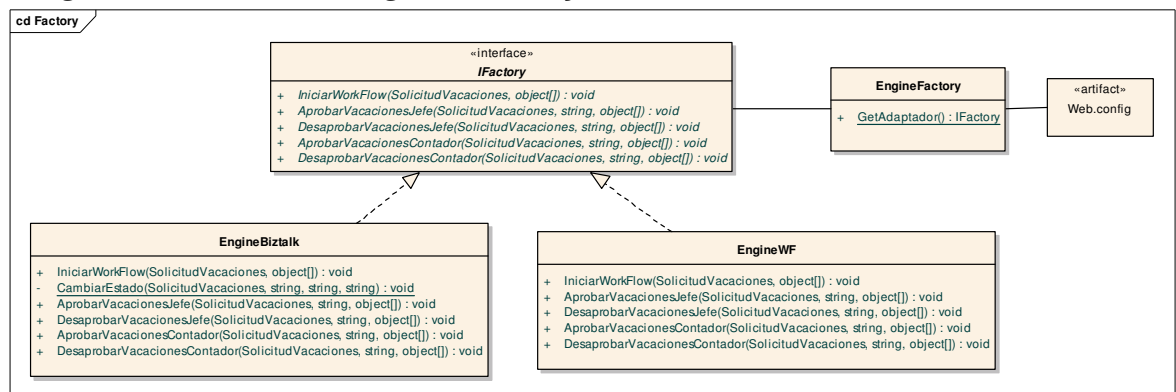


Figura 8. Diagrama de clases Engine Factory

PAQUETES DE DISEÑO ARQUITECTURALMENTE SIGNIFICATIVOS

Descripción

La aplicación se divide en 2 paquetes físicos.

Proyecto Web: En este paquete estará toda la interfaz de la aplicación a su vez los controladores, se usará el EngineFactory para instanciar una clase concreta de los motores de ejecución como EngineBiztalk o EngineWF, esta clase concreta se encuentra en la LibVacaciones en el paquete Factory.

Proyecto LibVacaciones: Es una librería de clases implementada en C# la cual se encapsula todos los requerimientos del proyecto, esta clase expone los servicios requeridos por los flujos de trabajo implementados en Windows Workflow Foundation y Biztalk Server.

Este proyecto se compone de los siguientes paquetes:

- **Facade:** Es la clase que expone todos los métodos necesarios para ser ejecutados por los Servicios Web.
- **Factory:** Este paquete es donde se implementa el patrón factory, el cual en tiempo de ejecución decide cual motor de procesos de negocio ejecutar.
- **BusinessLogic:** Es la capa de lógica de negocio donde se realizan las validaciones, esta interactúa directamente con la capa de acceso a datos.
- **BusinessEntities:** Son las clases del negocio.
- **Dao:** Son las clases encargadas de interactuar directamente con el motor de base de datos relacional SQL Server 2005.

Servicios Web: Son los servicios que consumen la funcionalidad implementada en la librería LibVacaciones, estos servicios serán consumidos por el proyecto web.

- **WsNomina:** Servicio Web que trae la información de los empleados.
- **WsEngine:** Servicio Web encargado de iniciar el motor de ejecución del proceso de negocio, ya sea Biztalk o Windows Workflow Foundation.
- **WsVacaciones:** Servicio Web que expone toda la funcionalidad de ingreso de solicitudes, cambio de estado, este servicio es consumido por el proyecto web y por los motores de ejecución de procesos de negocio.
- **WsNotificaciones:** Servicio Web que expone la funcionalidad para enviar correos electrónicos y enviar mensajes SMS por el sistema interno de la compañía Jabber, este servicio es consumido por los motores de proceso de negocio al ingresar solicitudes o cambiar el estado de ellas.

Nota: El controlador consume también directamente el **EngineFactory** para poder iniciar el motor de Windows Workflow Foundation, debido a que los objetos que utiliza no son serializables y no pueden ser expuestos mediante servicios web.

DIAGRAMA DE PAQUETES

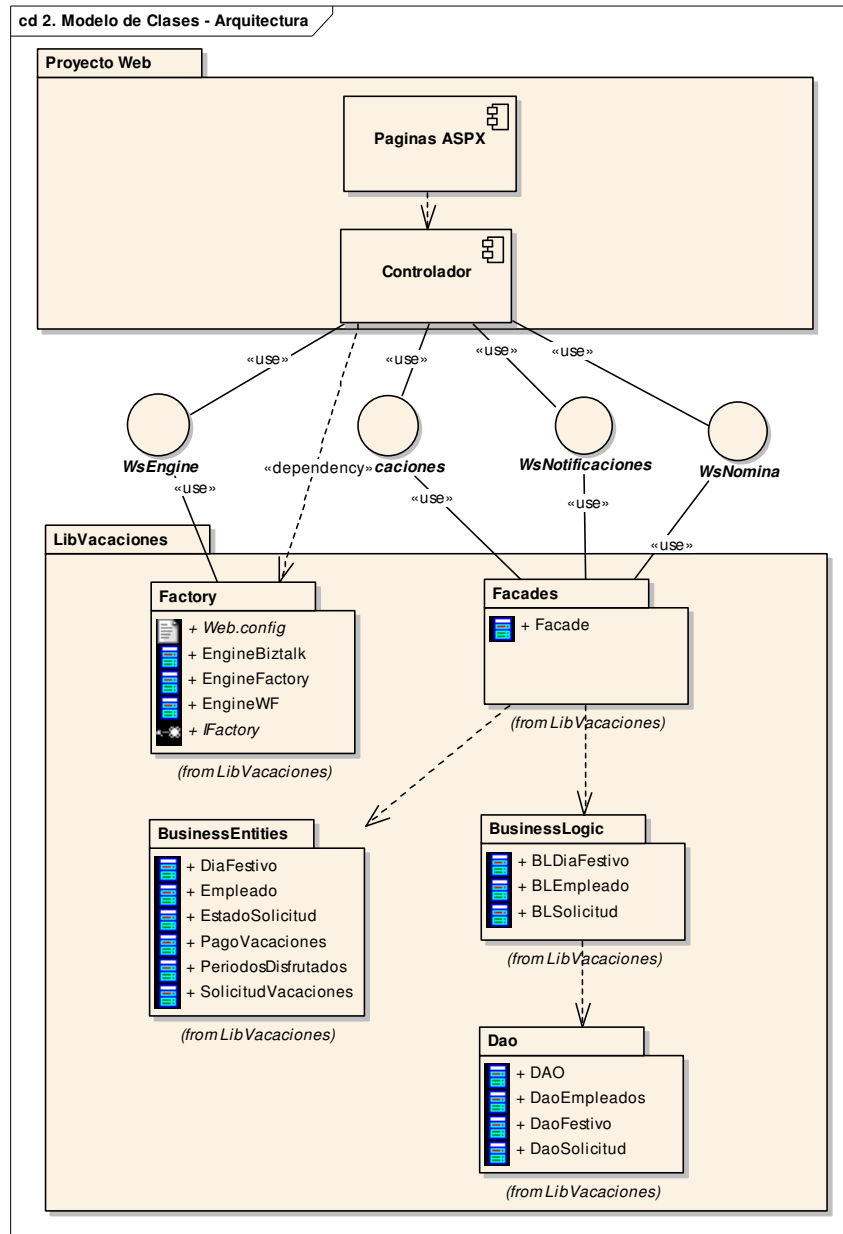


Figura 9. Diagrama de paquetes

VISTA DE PROCESOS

DIAGRAMA DE SECUENCIA – CREACIÓN DEL MOTOR

Descripción

El motor de ejecución del proceso de solicitud de vacaciones debe ser configurable, para esto en el archivo de configuración se pone la siguiente línea.

```
<add key="AdaptadorClass"  
value="Avansoft.Vacaciones.LibVacaciones.Factory.EngineWF"/>
```

El usuario realiza la acción de ingresar una nueva solicitud, cuando esto pasa se manda un mensaje al **Controlador**, que consulta mediante el método **GetFactory** de la clase **EngineFactory** cuál es el motor de ejecución configurado, una vez se tiene el motor Instanciado (en el diagrama se ve **EngineWorkFlow** pero también puede ser **EngineBiztalk**), se procede a llamar al método **IniciarWorkFlow**.

Este mismo diagrama también aplica para las siguientes acciones:

- Aprobar solicitud jefe inmediato
- Desaprobar solicitud jefe inmediato
- Aprobar solicitud contador
- Desaprobar solicitud contador

Diagrama

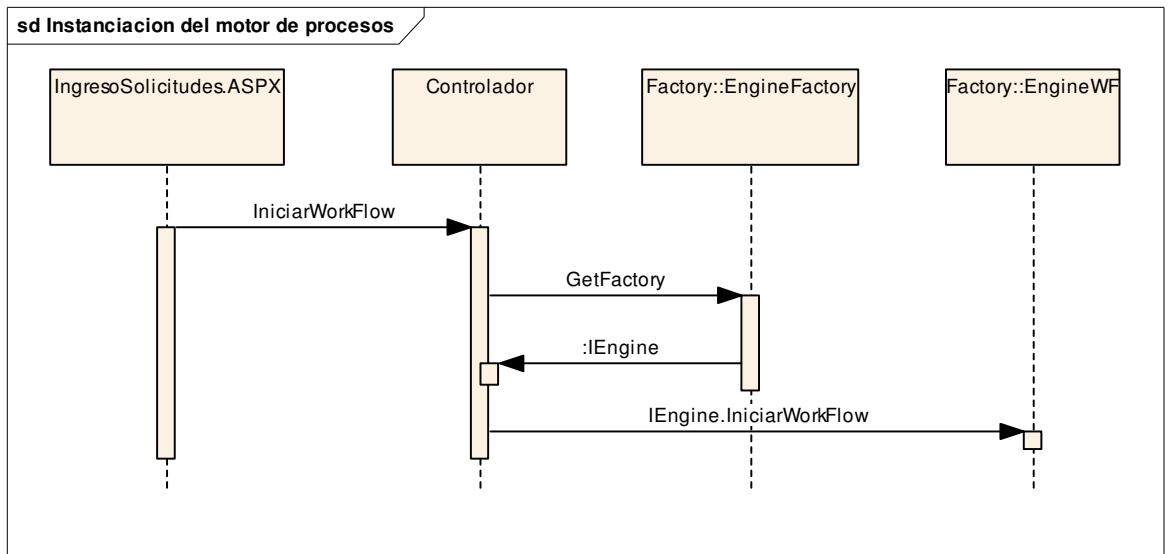


Figura 10. Diagrama de secuencia de instanciación del motor de procesos

VISTA DE COMPONENTES

Descripción

Los componentes implementados en el sistema son:

Proyecto Web: Implementa toda la capa de presentación.

LibVacaciones: Implementa toda la lógica de negocio y del motor del proceso de negocios configurado.

Servicios Web:

Implementa los servicios web de nómina, vacaciones y de notificaciones los cuales son usados por los flujos de trabajo realizados en Windows Workflow Foundation y Biztalk Server.

DIAGRAMA DE COMPONENTES

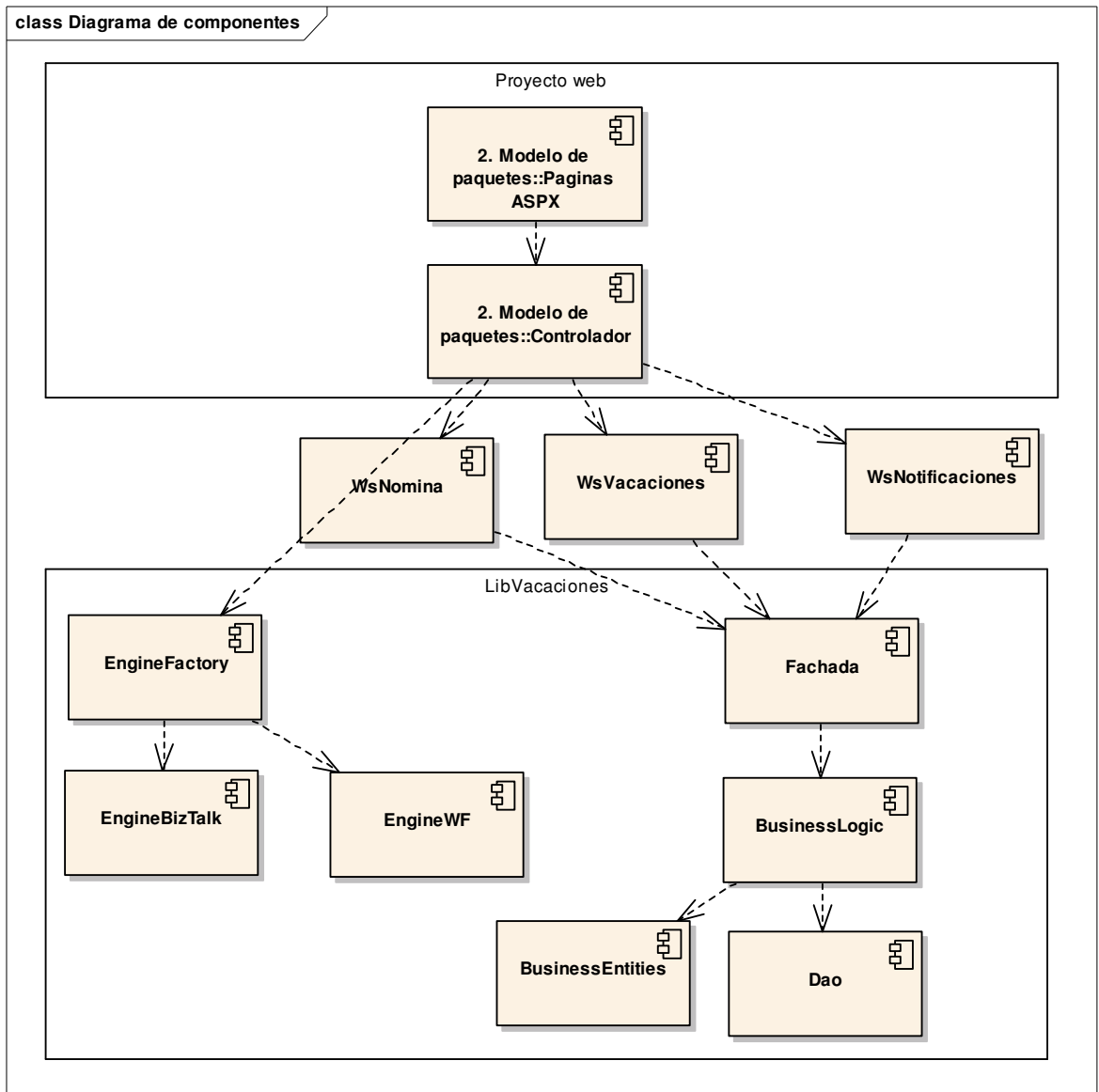


Figura 11. Diagrama de componentes

VISTA DE DESPLIEGUE

DESCRIPCIÓN DE LOS NODOS.

Servidor Internet Information Server 1: Este servidor alojará la aplicación web que será la interfaz de usuario final.

Servidor Internet Information Server 2: Este servidor alojará la **LibVacaciones**, librería que expone los servicios **WsNomina**, **WsNotificaciones** y **WsVacaciones**, estos servicios son consumidos por la aplicación web para mostrar por ejemplo, la lista de vacaciones solicitadas.

En este servidor también se encuentra el **Engine**, este es el componente encargado de instanciar la clase concreta según explicado en el diagrama de secuencia para poder empezar los flujos de trabajo según el motor de procesos configurado.

Biztalk Server: en este servidor se encuentra instalado el Biztalk Server, y aquí se encuentra instalado el Proceso de Vacaciones con orquestaciones.

DIAGRAMA DE DESPLIEGUE

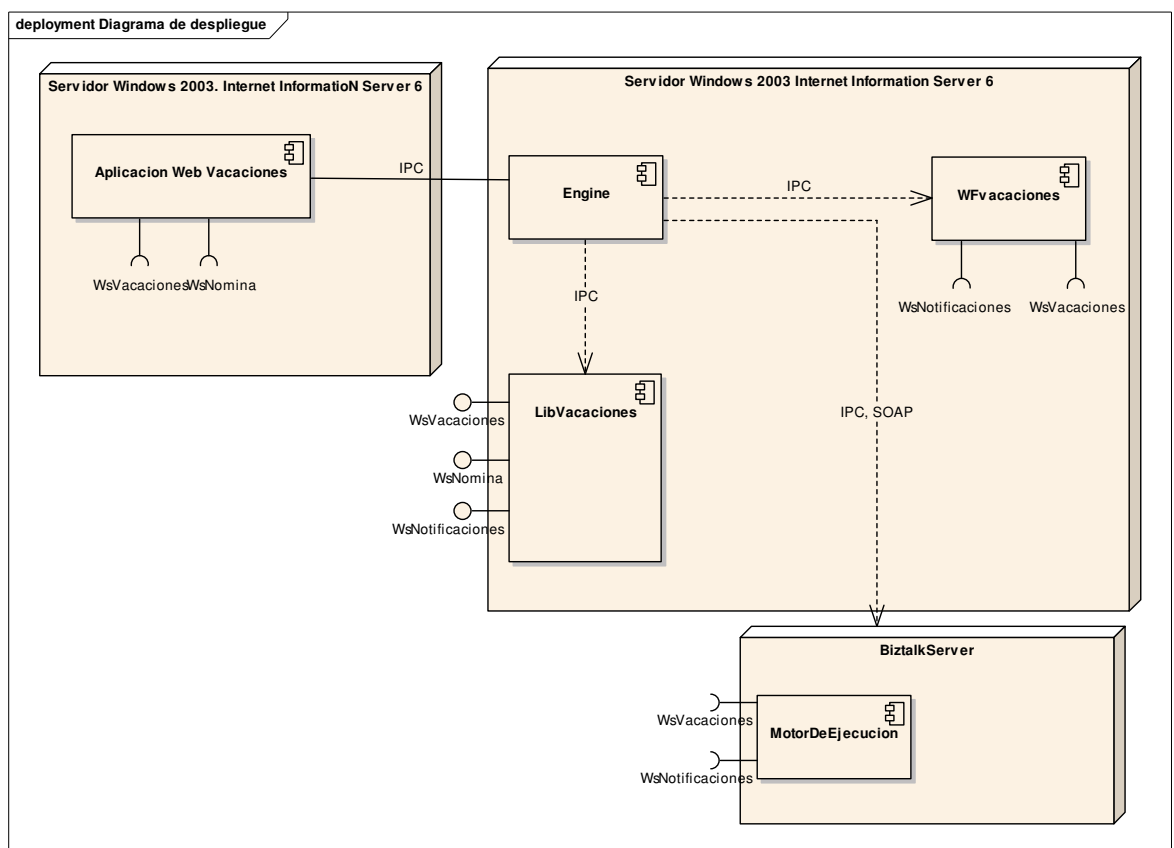


Figura 12. Diagrama de despliegue

INTERFACES

WSNOMINA

Nombre	WsNomina
Versión	1.0
Owner	David Kepes
Tipo	Negocio
Responsable	David Kepes
Accountable	David Kepes
Consulted	David Arango
Informed	Luis Fernando Londoño
Funcionalidad	<p>Requisitos Funcionales</p> <p>Provee los métodos necesarios para las operaciones básicas con la nómina de Avansoft.</p> <ul style="list-style-type: none">• Listado de empleados.• Cálculo de días hábiles para tomar vacaciones.• Cálculo de días festivos. <p>Service Operations:</p> <ul style="list-style-type: none">• List<Empleado> ListadoEmpleados.• Empleado GetEmpleado(int cedula)• Empleado GetEmpleado(string strUsuario)• Int CalculoDiasHabiles(DateTime fInicial, DateTime fFinal)• Int CalculoDiasTotales

	(DateTime fInicial, DateTime, fFinal)
Protocolo	SOAP
Url	
Requisitos no funcionales	<p>Security Constraints: El servicio puede ser invocado sólo por el usuario especial de ejecución de webservice ASPSYS.</p> <p>Quality of Service: El horario de función del webservice debe ser como mínimo el horario laboral de Avansoft: Lunes a Viernes 7.:30 AM a 5.30PM</p> <p>Transactional: No</p> <p>Service Level Agreement: 5 segundos.</p>

Tabla 3. Contrato del servicio WsNomina

WSVACACIONES

Nombre	WsVacaciones
Versión	1.0
Owner	David Kepes
Tipo	Negocio

Responsible	David Kepes
Accountable	David Kepes
Consulted	David Arango
Informed	Luis Fernando Londoño
Funcionalidad	<p>Requisitos Funcionales</p> <p>Provee los métodos necesarios para las operaciones que manipulan la información de vacaciones de los empleados de Avansoft.</p> <ul style="list-style-type: none"> • Gestionar el proceso de solicitud de vacaciones. <p>Service Operations:</p> <ul style="list-style-type: none"> • Int GetDiasDisponibles(int periodo) • Int GetDiasDisfrutados(int periodo) • IngresarSolicitudVacaciones(int cedula, DateTime fInicial, DateTime fFinal, string PagoVacaciones) • AprobarVacacionesDirector(int idSolicitud, string strDirector) • DesaprobarVacacionesDirector(int idSolicitud, string strDirector) • AprobarVacacionesContador(int idSolicitud, string strContador) • DesAprobarVacacionesContador(int idSolicitud, string strContador) • List<Solicitudes> GetSolicitudesDirector(string strdirector) • List<Solicitudes>

	GetSolicitudesContador(string strdirector)
Protocolo	SOAP
Url	
Requisitos no funcionales	<p>Security Constraints: El servicio puede ser invocado sólo por el usuario especial de ejecución de webservice ASPSYS.</p> <p>Quality of Service: El horario de función del webservice debe ser como mínimo el horario laboral de Avansoft: Lunes a Viernes 7.:30 AM a 5.30PM</p> <p>Transactional: No</p> <p>Service Level Agreement: 6 segundos.</p>

Tabla 4. Contrato del servicio WsVacaciones

WSNOTIFICACIONES

Nombre	WsNotificaciones
Versión	1.0
Owner	David Kepes
Tipo	David Kepes

Responsible	David Kepes
Accountable	David Kepes
Consulted	David Kepes
Informed	Andrés Gómez
Funcionalidad	<p>Requisitos Funcionales</p> <p>Provee los métodos necesarios para las operaciones que manipulan las notificaciones por correo electrónico u otros medios.</p> <p>Service Operations:</p> <ul style="list-style-type: none"> • EnviarCorreo(string strTo, string strSubject, string strBody) • EnviarNotificacionSpark(string strTo, string strMensaje)
Protocolo	SOAP
Url	
Requisitos no funcionales	<p>Security Constraints:</p> <p>El servicio puede ser invocado sólo por el usuario especial de ejecución de webservice ASPSYS.</p> <p>Quality of Service:</p> <p>El horario de función del webservice debe ser como mínimo el horario laboral de Avansoft:</p> <p>Lunes a Viernes 7.:30 AM a 5.30PM</p> <p>Transactional:</p> <p>No</p>

	Service Level Agreement: 7 segundos.
--	---

Tabla 5. Contrato del servicio WsNotificaciones

PLATAFORMA TECNOLÓGICA

Particionamiento de Aplicación		Vista	Servicios	Dominio		
Capas de Plataforma	Cliente	Presentación	Negocio	Integración	Recursos	
Aplicación	HTML	Web Forms ASP.NET 2.0 Patron Business Delegate ASP.NET AJAX ASP.NET 1.0 Patron Service Locator	Patrón Fachada	Patron DAO ADO.NET Patron Factory Patron Service Locator	Modelo de Datos E/R Procedimientos Almacenados	USABILIDAD ESCALABILIDAD, DESEMPEÑO, SEGURIDAD, MANTENIBILIDAD,
Plataforma Virtual	HTML 4 o sup	.NET Framework 2,0	.NET Framework 2.0		ANSI-SQL	
Plataforma Superior	Internet Explorer 7.0	IIS 6,0	IIS .NET Framework 2.0	IIS .NET Framework 2.0, .NET FRAMEWORK 3.0, BIZTALK SERVER	SQL Server 2005	
Plataforma Inferior	Windows XP	Windows 2003 Server	IIS 6,0	IIS 6,0	Windows 2003 Server	
Plataforma de Red	TCP	TCP	TCP	TCP	TCP	
Plataforma de Hardware	P4 3.0Ghz, 1GB RAM	P4 3.0Ghz, 2GB RAM	P4 3.0Ghz, 2GB RAM	P4 3.0Ghz, 2GB RAM	P4 3.0Ghz, 2GB RAM	

IMPLEMENTACION

La infraestructura de red que se va a utilizar es presentada en el diagrama siguiente:

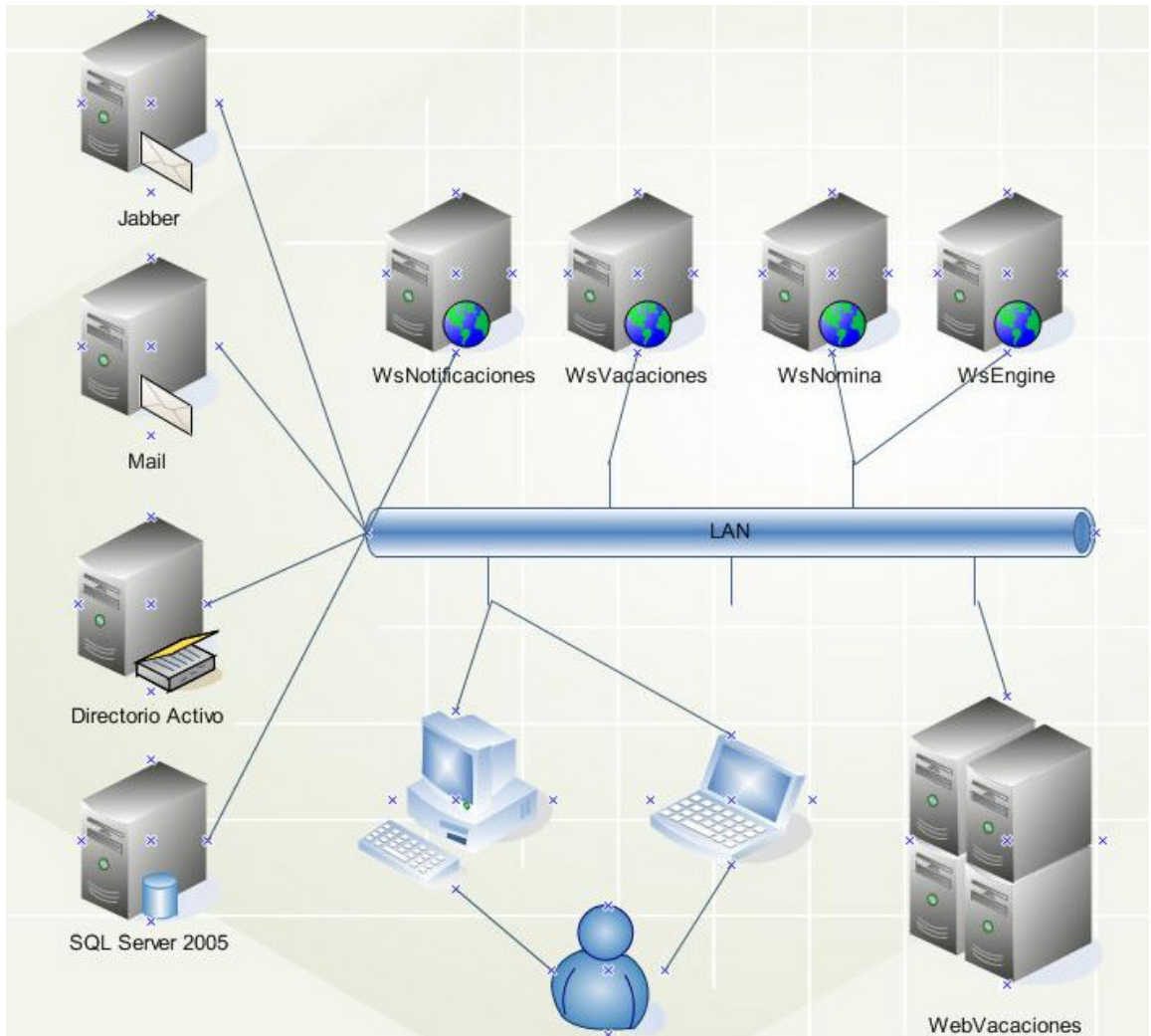


Figura 13. Diagrama de infraestructura de red y servidores

Para la implementación se utilizarán los servicios web disponibles en la red y los recursos ofrecidos por el servidor de Directorio activo, SQL Server 2005, Mail y Jabber.

La implementación parte del proceso de autenticación, que se produce contra el directorio activo, luego la autorización es realizada contra el servicio web WsNomina quien devuelve el listado de los empleados, el cual entre uno de los campos de retorno incluye el cargo.

La aplicación conociendo el cargo del empleado despliega las siguientes opciones en el menú principal:

Analista:

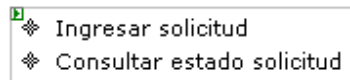


Figura 14. Menú del analista

Director:

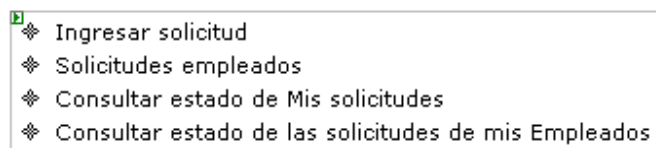


Figura 15. Menú del director

Contador:

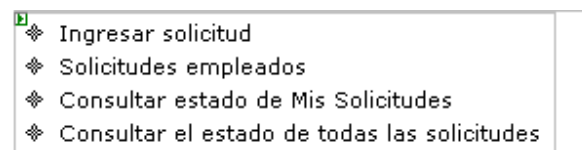


Figura 16. Menú del contador

Debido a que es necesario contar con un mecanismo que permita cambiar el motor de ejecución de procesos de negocio, en este caso específico para el proceso de negocio de solicitud de vacaciones, se vio la oportunidad de implementar el patrón de diseño **Factory**, para esto la interfaz debe contar con los métodos que permitan activar la ejecución del flujo de trabajo y las clases concretas deben implementar la funcionalidad que corresponda con la especificación de cada motor para dicho proceso de activación.

La interfaz actualmente se encuentra de la siguiente manera:

```

using System;
using System.Collections.Generic;
using System.Text;
using Avansoft.Vacaciones.LibVacaciones.BusinessEntities;

namespace Avansoft.Vacaciones.LibVacaciones.Factory
{
    public interface IFactory
    {
        void IniciarWorkFlow(SolicitudVacaciones objSolicitud, object[] args);
        void AprobarVacacionesJefe(SolicitudVacaciones objSolicitud, string
strComentarios, object[] args);
        void DesaprobarVacacionesJefe(SolicitudVacaciones objSolicitud, string
strComentarios, object[] args);
        void AprobarVacacionesContador(SolicitudVacaciones objSolicitud, string
strComentarios, object[] args);
        void DesaprobarVacacionesContador(SolicitudVacaciones objSolicitud,
string strComentarios, object[] args);
    }
}

```

Código 1. Interfaz IFactory

La implementación se verá más adelante cuando se aborde cada caso concreto con la tecnología específica.

La opción de ingresar solicitud de vacaciones es la encargada de iniciar el flujo de trabajo según la configuración elegida en el **web.config** y se hace de la siguiente forma obtenido los datos de la pantalla de ingreso de solicitud.

```

IFactory adp = EngineFactory.GetAdaptador();
SolicitudVacaciones objSol = new SolicitudVacaciones();
objSol.DtmFechaInicial = Convert.ToDateTime(TxtFechaInicial.Text);
objSol.DtmFechaFinal = Convert.ToDateTime(TxtFechaFinal.Text);
objSol.PagoVacaciones =
SolicitudVacaciones.GetPago(rdlPagoVacaciones.SelectedValue, "");
objSol.Empleado =
Facade.GetInstance().GetEmpleadoxUsuario(txtUsuario.Text);
objSol.Dias = int.Parse(LblDiasTotales.Text);
adp.IniciarWorkFlow(objSol, new object[] { Application["WorkflowRuntime"]
});

```

Código 2. Iniciar WorkFlow

El método IniciarWorkFlow de la interfaz IFactory es el encargado dependiendo de la implementación, bien sea EngineBiztalk o EngineWF, de activar el flujo de trabajo según la implementación que se muestra en los siguientes dos puntos.

Las siguientes opciones también requieren la interacción del usuario para permitir al flujo continuar con su ejecución:

Aprobar vacaciones jefe inmediato: Esta opción sólo se presenta al jefe inmediato que tenga personas a cargo con alguna solicitud pendiente.

```
IFactory objAdaptador = EngineFactory.GetAdaptador();
int idSolicitud = int.Parse(lblIdsolicitud.Text.ToString());
SolicitudVacaciones objSolicitud =
Facade.GetInstance().GetSolicitud(idSolicitud);
objAdaptador.AprobarVacacionesJefe(objSolicitud, txtcomentarios.Text,
new object[] { Application["WorkflowRuntime"], Application["VacacionesService"]
});
```

Código 3. Aprobar vacaciones jefe

Desaprobar vacaciones jefe inmediato: Esta opción sólo se presenta al jefe inmediato que tenga personas a cargo con alguna solicitud pendiente.

```
IFactory objAdaptador = EngineFactory.GetAdaptador();
int idSolicitud = int.Parse(lblIdsolicitud.Text.ToString());
SolicitudVacaciones objSolicitud =
Facade.GetInstance().GetSolicitud(idSolicitud);
objAdaptador.DesaprobarVacacionesJefe(objSolicitud, txtcomentarios.Text,
new object[] { Application["WorkflowRuntime"], Application["VacacionesService"]
});
```

Código 4. Desaprobar vacaciones jefe

Aprobar vacaciones contador: Esta opción sólo se presenta al Contador y muestra todas las solicitudes pendientes que han sido aprobadas por un director.

```
IFactory objAdaptador = EngineFactory.GetAdaptador();
int idSolicitud = int.Parse(lblIdsolicitud.Text.ToString());
```

```

    SolicitudVacaciones objSolicitud =
Facade.GetInstance().GetSolicitud(idSolicitud);
    objAdaptador.AprobarVacacionesContador(objSolicitud,
txtcomentarios.Text, new object[] { Application["WorkflowRuntime"],
Application["VacacionesService" ]});

```

Código 5. Aprobar vacaciones contador

Desaprobar vacaciones contador: Esta opción sólo se presenta al Contador y muestra todas las solicitudes pendientes que han sido aprobadas por un director.

```

IFactory objAdaptador = EngineFactory.GetAdaptador();
int idSolicitud = int.Parse(lblIdsolicitud.Text.ToString());
SolicitudVacaciones objSolicitud =
Facade.GetInstance().GetSolicitud(idSolicitud);
    objAdaptador.DesaprobarVacacionesContador(objSolicitud,
txtcomentarios.Text, new object[] { Application["WorkflowRuntime"],
Application["VacacionesService" ]});

```

Código 6. Desaprobar vacaciones contador

IMPLEMENTACIÓN EN WINDOWS WORKFLOW FOUNDATION

La implementación del proceso de negocio en Windows Workflow Foundation se llevó a cabo utilizando solamente un flujo de trabajo de tipo Secuencial Workflow. Las actividades que fueron adicionas al flujo son las siguientes:

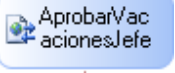

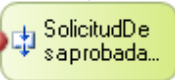
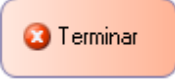
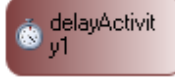
Actividad	Nombre	Descripción
 AprobarVacacionesJefe	InvokeWebService	Permite invocar un servicio web
 listenActivity1	ListenActivity	Permite detener el flujo esperando a que un evento ocurra
 SolicitudDe s aprobada...	HandleExternalEvent	Permite escuchar por un evento específico
 Terminar	Terminate	Termina el flujo de trabajo
 delayActivit y1	Delay	Permite detener el flujo durante un periodo en el tiempo

Tabla 6. Actividades de WorkFlow Foundation

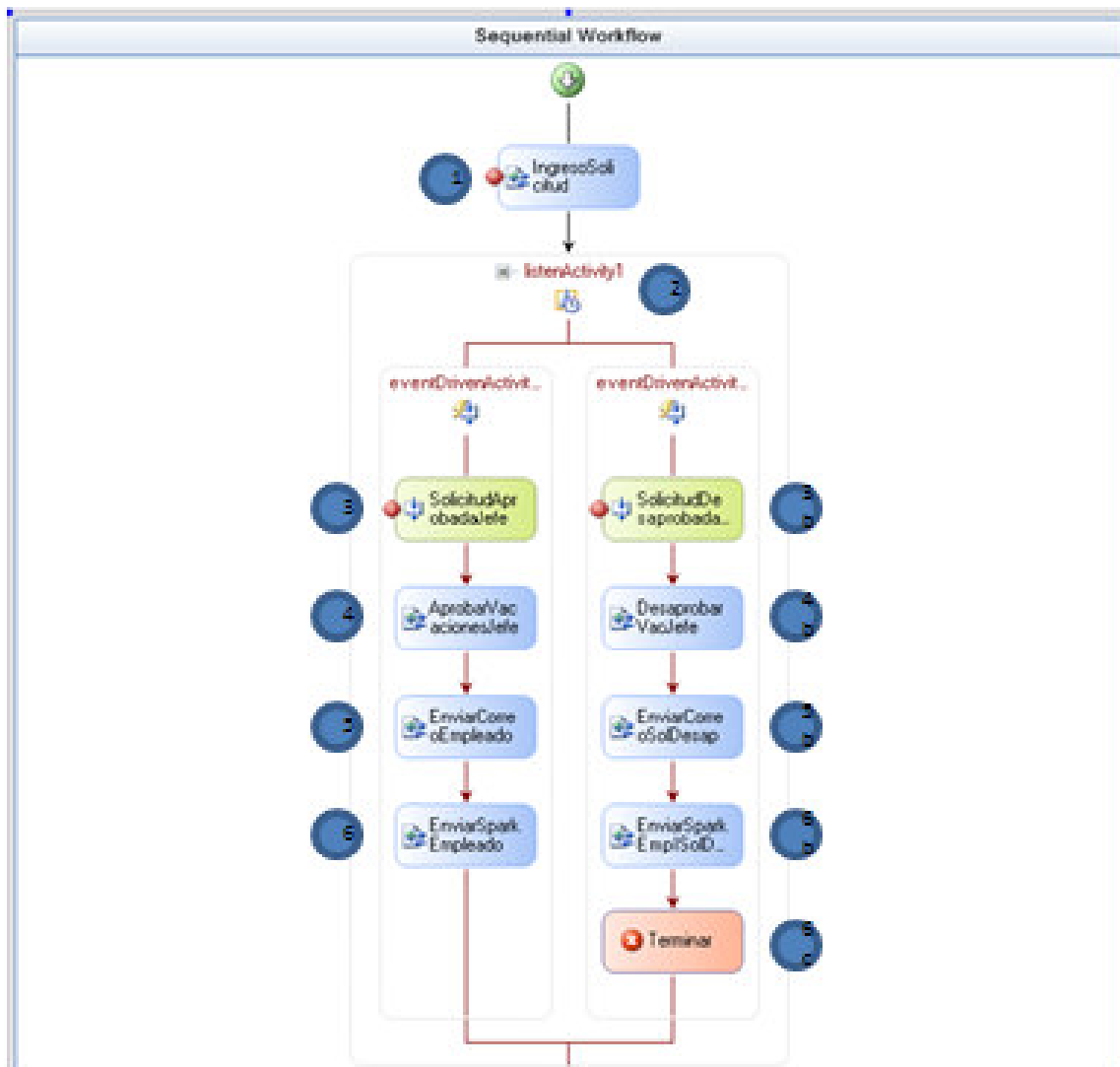


Figura 17. Diagrama de flujo de trabajo Workflow Foundation 1

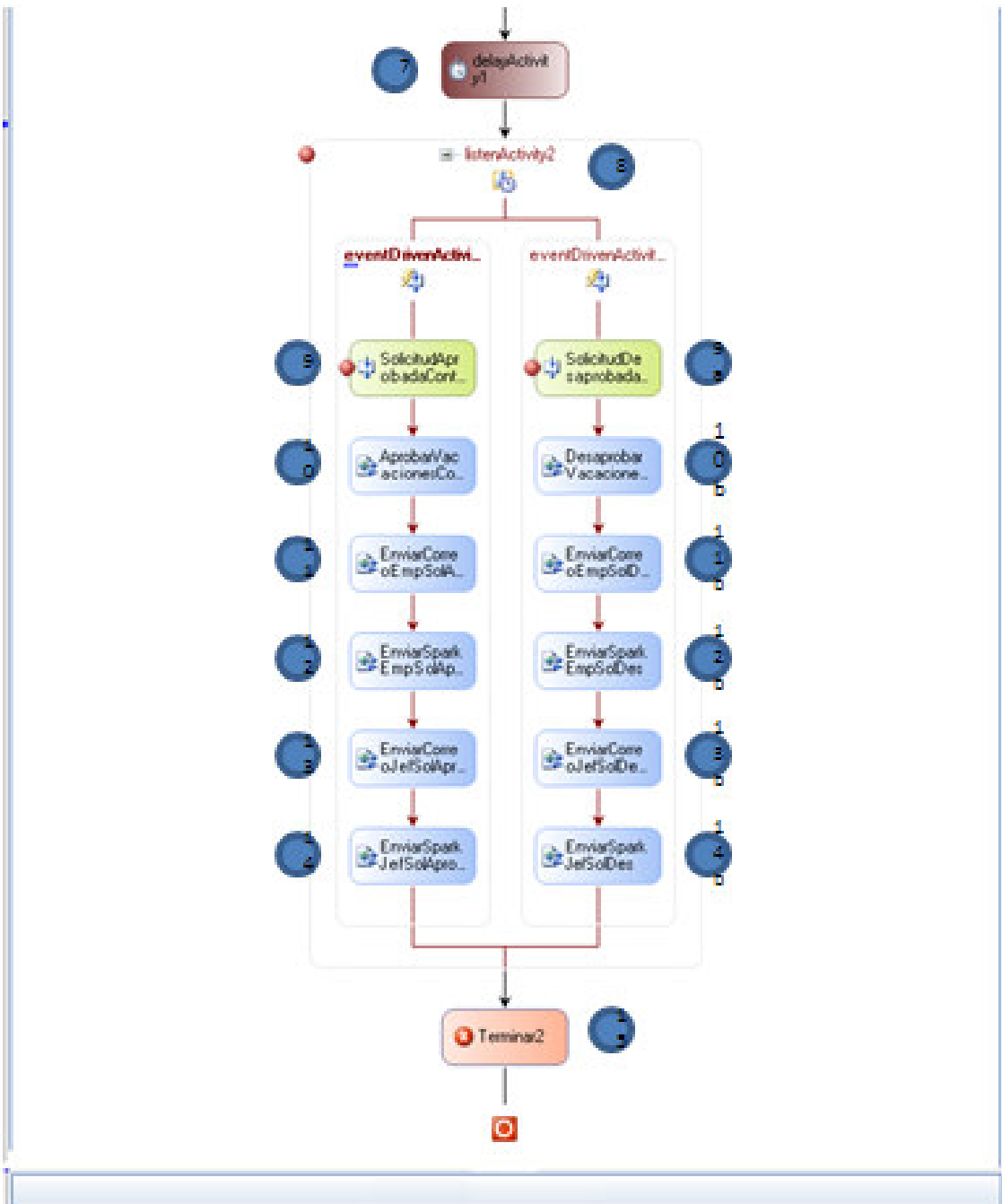


Figura 18. Diagrama de flujo de trabajo Workflow Foundation 2

El flujo de trabajo es activado por el método **IniciarWorkflow** de la clase **EngineWF** la cual implementa la interfaz **IFactory** mencionada anteriormente, dicha implementación pasa los parámetros al Workflow e inicia el flujo de la siguiente manera:

```
public void IniciarWorkFlow(SolicitudVacaciones objSolicitud, object[] args)
{
    WorkflowRuntime workflowRuntime = (WorkflowRuntime)args[0];
    Dictionary<string, object> parameters = new Dictionary<string, object>();
    parameters.Add("strUsuario", objSolicitud.Empleado.StrUsuario);
    parameters.Add("dteFechaIni", objSolicitud.DtmFechaInicial);
    parameters.Add("dteFechaFinal", objSolicitud.DtmFechaFinal);
    parameters.Add("objPagoVacaciones",
        objSolicitud.GetPago(objSolicitud.PagoVacaciones));
    parameters.Add("StrCorreoContador",
        ConfigurationManager.AppSettings["StrCorreoContador"]);
    if (objSolicitud.Empleado.JefeInmediato != null)
    {
        parameters.Add("StrCorreoJefe",
objSolicitud.Empleado.JefeInmediato.StrEmail);
    }
    parameters.Add("StrCorreoEmpleado",objSolicitud.Empleado.StrEmail);

    parameters.Add("StrAsunto",ConfigurationManager.AppSettings["StrAsunto"]);
    parameters.Add("StrMensajeSolAprobadaJefe",
        ConfigurationManager.AppSettings["StrMensajeSolAprobadaJefe"]);
    parameters.Add("StrMensajeSolDesaprobadaJefe",
        ConfigurationManager.AppSettings["StrMensajeSolDesaprobadaJefe"]);
    parameters.Add("StrMensajeSolAprobadaContador",
ConfigurationManager.AppSettings["StrMensajeSolAprobadaContador"]);
    parameters.Add("StrMensajeSolDesaprobadaContador",
ConfigurationManager.AppSettings["StrMensajeSolDesaprobadaContador"]);
    parameters.Add("StrMensajeSolIngresada",
        ConfigurationManager.AppSettings["StrMensajeSolIngresada"]);
    WorkflowInstance instance = workflowRuntime.CreateWorkflow(
        typeof(Avansoft.Vacaciones.WFVacaciones.WFVacaciones),
parameters);
    instance.Start();
}
```


Código 7. Implementación de IniciarWorkflow

Para poder iniciar correctamente el flujo de trabajo es necesario contar con el WorkflowRuntime correctamente instanciado puesto que es el encargado de crear, alojar y administrar todo lo relacionado a las WorkFlow instances.

Debido a que la instancia del WorkFlowRuntime debe permanecer durante todo el tiempo de la aplicación se decidió incluirla en una variable de Application que tiene ámbito para toda la aplicación y todos los usuarios. La instancia del WorkflowRuntime es creada en el método Application_Start el cual se ejecuta una sola vez cuando la aplicación es iniciada, y es allí donde se adiciona el servicio de persistencia al WorkFlowRuntime SqlWorkflowPersistenceService quien es el que se encarga de mantener una versión de la instancia del WorkFlow serializada en la base de datos. La instancia es guardada en el momento en que el Workflow pasa a un estado de espera, por ejemplo, cuando encuentra una actividad de tipo ListenActivity, y es conservada utilizando el GuidInstance con el que es posible recuperarla y seguir trabajando con ella más adelante, a continuación se muestra el método Application_Start del Global.asax:

```
void Application_Start(object sender, EventArgs e)
{
    System.Workflow.Runtime.WorkflowRuntime workflowRuntime =
    new System.Workflow.Runtime.WorkflowRuntime();
    string connectionString = @"Persist Security Info=False;User
ID=sa;password=12345;Initial Catalog=DBVACACIONES;Data
Source=david\sqlexpress";
    bool unloadOnIdle = true;
    TimeSpan instanceOwnershipDuration = TimeSpan.MaxValue;
    TimeSpan loadingInterval = new TimeSpan(0, 2, 0);
    Application["loadingInterval"] = loadingInterval;
    Application["instanceOwnershipDuration"] = instanceOwnershipDuration;
    NameValueCollection objCol = new NameValueCollection();
    objCol.Add("unloadOnIdle", "true");
    objCol.Add("connectionString", connectionString);
}
```

```

        System.Workflow.Runtime.Hosting.SqlWorkflowPersistenceService
persistService = new
System.Workflow.Runtime.Hosting.SqlWorkflowPersistenceService(objCol);

        Avansoft.Vacaciones.WFVacaciones.WFVacacionesEvents
VacacionesService = new
Avansoft.Vacaciones.WFVacaciones.WFVacacionesEvents();
        System.Workflow.Activities.ExternalDataExchangeService dataService =
new System.Workflow.Activities.ExternalDataExchangeService();
        workflowRuntime.AddService(dataService);
        dataService.AddService(VacacionesService);

        workflowRuntime.AddService(persistService);
        workflowRuntime.StartRuntime();

        Application["WorkflowRuntime"] = workflowRuntime;
        Application["VacacionesService"] = VacacionesService;

    }

```

Código 8. Application Start en el Global.asax

Luego que el workflow es iniciado comienza su ejecución de la siguiente manera:

1. Se llama al web method **IngresarSolicitud** del web service **WsVacaciones**.
2. El flujo se persiste y espera por un evento externo. Para la implementación de los eventos externos se utiliza la siguiente interfaz que es la que se asocia a las actividades HandleExternalEvent

```

[ExternalDataExchange]
public interface IWFVacaciones
{
    event EventHandler<SolicitudVacacionesArgs> SolicitudAprobadaJefe;
    event EventHandler<SolicitudVacacionesArgs> SolicitudDesaprobadaJefe;
    event EventHandler<SolicitudVacacionesArgs> SolicitudAprobadaContador;
    event EventHandler<SolicitudVacacionesArgs>
SolicitudDesaprobadaContador;
}

```

Código 9. Interfaz IWFVacaciones para eventos del workflow

Debido a que el evento `SolicitudAprobadaJefe` es lanzado por una interacción humana de un jefe inmediato con la aplicación web el método `AprobarVacacionesJefe` de la interfaz `IFactory` debe ser llamado por la aplicación y la clase que implementa el método para el motor de `WorkFlow` es la encargada de elevar el evento `SolicitudAprobadaJefe` utilizando la clase concreta que implementa la interfaz, a continuación se presenta el método `AprobarVacacionesJefe` de la clase `EngineWF`, seguido por la implementación del método `AprobarSolicitudJefe` y `DesaprobarSolicitudJefe` de la clase `WfVacacionesEvents`:

```
public void AprobarVacacionesJefe(SolicitudVacaciones objSolicitud,string
strComentarios, object[] args)
{
    WorkflowRuntime workflowRuntime = (WorkflowRuntime)args[0];
    WfVacacionesEvents objWfVacEvents = (WfVacacionesEvents)args[1];

    objWfVacEvents.instanceId =
workflowRuntime.GetWorkflow(objSolicitud.WfInstance);
    objWfVacEvents.SolicitudId = objSolicitud.IdSolicitud;
    objWfVacEvents.AprobarSolicitudJefe(strComentarios);
}
```

Código 10. Implementación de `AprobarVacacionesJefe`

```
public void AprobarSolicitudJefe(string strComentarios)
{
    EventHandler<SolicitudVacacionesArgs> solAprobadaJefe =
this.SolicitudAprobadaJefe;
    if (solAprobadaJefe != null)
        solAprobadaJefe(null, new
SolicitudVacacionesArgs(instanceId.InstanceId, SolicitudId, strComentarios));
}

public void DesaprobarSolicitudJefe(string strComentarios)
{
    EventHandler<SolicitudVacacionesArgs> solDesaprobadaJefe =
this.SolicitudDesaprobadaJefe;
```

```
        if (solDesaprobadaJefe != null)
            solDesaprobadaJefe(null, new
SolicitudVacacionesArgs(instanceld.Instanceld, SolicitudId, strComentarios));
    }
```

Código 11. Evento AprobarSolicitudJefe

Al elevar el evento el flujo continúa su ejecución de la siguiente manera:

3. Se llama al web method **AprobarVacacionesJefe** del web service **WsVacaciones**.
 - b. En caso que el evento haya sido **DesaprobarSolicitudJefe** se llama al web method **DesaprobarVacacionesJefe** del web service **WsVacaciones**.
4. Se envía el correo al empleado con el mensaje de aprobación de las vacaciones
 - b. En caso que las vacaciones hayan sido desaprobadas se envía un correo al empleado con el mensaje de desaprobación de las vacaciones.
5. Se envía un mensaje al empleado utilizando al servidor de jabber, el mensajero interno de la compañía diciendo que se aprueban las vacaciones
 - b. En caso que las vacaciones hayan sido desaprobadas, Se envía un mensaje al empleado utilizando al servidor de jabber, el mensajero interno de la compañía diciendo que se desaprueban las vacaciones
 - c. Termina el flujo de trabajo.
6. Se detiene el flujo por un momento con el fin de que se pueda dar un tiempo para utilizar el servicio de persistencia y guardar los datos en la base de datos.
7. Se espera por un evento de la misma manera que se vio en el paso 2.
8. Se espera a que ocurra el evento SolicitudAprobadaContador

- b. Se espera a que ocurra el evento `SolicitudDesaprobadaContador`.
- 9. Se llama al web method **AprobarVacacionesContador** del web service **WsVacaciones**.
 - b. En caso que el evento haya sido `SolicitudDesaprobadaContador` se llama al web method **DesaprobarVacacionesContador** del web service **WsVacaciones**.
- 10. Se envía el correo al empleado con el mensaje de aprobación de las vacaciones del contador
 - b. En caso que las vacaciones hayan sido desaprobadas se envía un correo al empleado con el mensaje de desaprobación de las vacaciones.
- 11. Se envía un mensaje al empleado utilizando al servidor de jabber, el mensajero interno de la compañía diciendo que se aprueban las vacaciones por parte del contador
 - b. En caso que las vacaciones hayan sido desaprobadas, Se envía un mensaje al empleado utilizando al servidor de jabber, el mensajero interno de la compañía diciendo que se desaprueban las vacaciones
- 12. Se envía el correo al jefe inmediato con el mensaje de aprobación de las vacaciones del contador
 - b. En caso que las vacaciones hayan sido desaprobadas se envía un correo al jefe inmediato con el mensaje de desaprobación de las vacaciones.
- 13. Se envía un mensaje al jefe inmediato utilizando al servidor de jabber, el mensajero interno de la compañía diciendo que se aprueban las vacaciones por parte del contador
 - b. En caso que las vacaciones hayan sido desaprobadas, Se envía un mensaje al jefe inmediato utilizando al servidor de jabber, el mensajero interno de la compañía diciendo que se desaprueban las vacaciones.

IMPLEMENTACIÓN EN BIZTALKSERVER

Como se ha hecho anotación en los diagramas de secuencia, componentes y de paquetes, la solución tiene una clase llamada **EngineBiztalk** que implementa la interfaz **IEngine** y mediante la cual se inician las orquestaciones de la solución.

En este caso los métodos implementados construyen un archivo XML en las carpetas destinadas para ellos según el archivo web.config.

Estos métodos son:

IniciarWorkFlow

AprobarVacacionesJefe

DesaprobarVacacionesJefe

AprobarVacacionesContador

DesaprobarVacacionesContador

Para la realización del flujo de trabajo en BizTalk Server se decidió implementar los mensajes en formato XML con el adaptador FILE que tiene BiztalkServer en su instalación básica. Este adaptador escucha ciertas carpetas configuradas en la solución, cuando detecta un archivo XML, BiztalkServer lee los datos de éste para iniciar el flujo de trabajo.

Se decidió partir la solución en 3 flujos básicos:

1. Ingresar Solicitud Empleado:

Este flujo de trabajo u Orquestación (concepto que usa Biztalk Server para los flujos), recibe un archivo XML en la carpeta configurada en el archivo web.config

```
<add key="folderEntrada" value="C:\ProcesoVacaciones\Entrada\"/>
```

Código 12. FolderEntrada en web.config

Esta carpeta debe tener permisos de lectura y escritura para el usuario que inicia los servicios de Biztalk, usualmente BIZTALK Service.

Los archivos que llegan a esta carpeta deben tener el siguiente esquema:

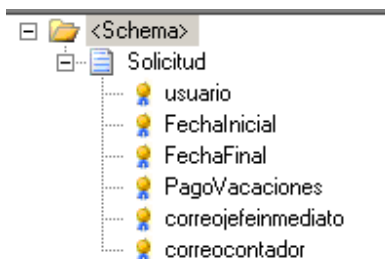


Figura 19. Esquema de archivos de solicitud vacaciones

```
<?XML version="1.0" ?>
<ns0:Solicitud xmlns:ns0="http://BTS_ProcesoVacaciones2.Solicitud">
  <usuario>levalencia</usuario>
  <FechaInicial>2007-11-19</FechaInicial>
  <FechaFinal>2007-11-11</FechaFinal>
  <PagoVacaciones>Nomina</PagoVacaciones>
  <correojefeinmediato>levalencia@avansoft.com</correojefeinmediato>
  <correocontador>dkepes@avansoft.com</correocontador>
</ns0:Solicitud>
```

Código 13. Ejemplo del esquema de archivos de solicitud vacaciones

Una vez el archivo es creado en la carpeta destino, el flujo de Biztalk corre de la siguiente manera:

- b. Se lee el archivo XML de la carpeta **FolderEntrada** configurada en el archivo web.config.
- c. Se construye un mensaje tipo **IngresarSolicitudRequest** para poder utilizarlo en la llamada al web service.

- d. Se llama al web method **IngresarSolicitud** del web service **WsVacaciones**.
- e. Se recibe la respuesta del web service.
- f. Se construye el mensaje **EnviarCorreoRequest** para enviar la notificación al jefe inmediato vía correo electrónico.
- g. Se llama al web method **EnviarCorreo** del web service **WsNotificaciones**.
- h. Se recibe la respuesta del web service.
- i. Se construye el mensaje **EnviarSparkRequest** para enviar la notificación al jefe inmediato vía Spark.
- j. Se llama el web method **EnviarSpark** del web service **WsNotificaciones**.
- k. Se recibe la respuesta del servicio.

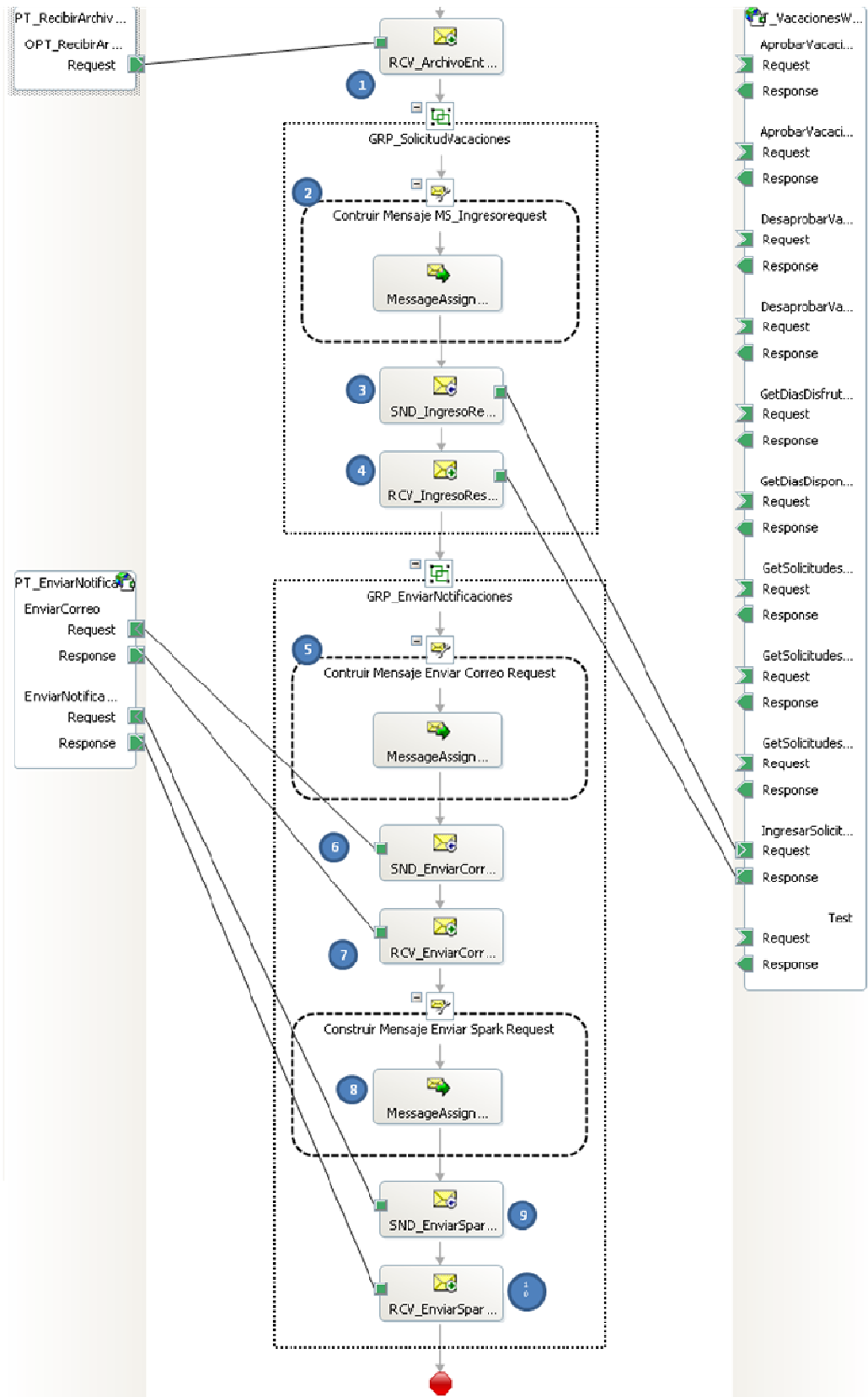


Figura 20. Diagrama del flujo de trabajo de ingresar solicitud en Biztalk

2. Cambio Estado Solicitud Jefe Inmediato:

Este flujo de trabajo recibe un archivo XML en la carpeta configurada en el archivo web.config con la clave **folderCambioEstadoJefe**.

```
<add key="folderCambioEstadoJefe"  
value="C:\ProcesoVacaciones\CambioEstadoJefe\"/>
```

Código 14. FolderCambioEstadoJefe en el web.config

Los archivos que llegan a esta carpeta deben tener el siguiente esquema:

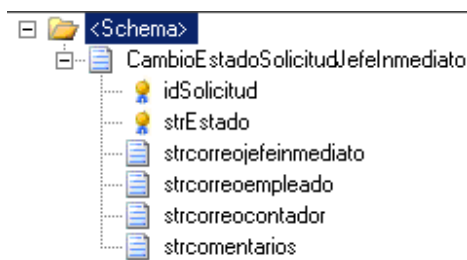


Figura 21. Esquema de archivos de cambio solicitud jefe

```
<?XML version="1.0" ?>  
  
<ns0:CambioEstadoSolicitudJefeInmediato  
xmlns:ns0="http://BTS_ProcesoVacaciones2.CambioEstadoSolicitud">  
  
  <idSolicitud>37</idSolicitud>  
  
  <strEstado>AprobadaJefe</strEstado>  
  
  <strcorreojefeinmediato>levalencia@avansoft.com</strcorreojefeinmediato>  
  
  <strcorreoempleado>dkepes@avansoft.com</strcorreoempleado>  
  
  <strcorreocontador>dkepes@avansoft.com</strcorreocontador>  
  
  <strcomentarios />
```

```
</ns0:CambioEstadoSolicitudJefeInmediato>
```

Código 15. Ejemplo del esquema de archivos para cambio de solicitud jefe

Una vez el archivo es creado en la carpeta destino, el flujo de Biztalk corre de la siguiente manera:

1. Se lee el archivo XML de la carpeta **folderCambioEstadoJefe** configurada en el archivo **web.config**.
2. Se extraen las variables que servirán durante el resto del flujo con el siguiente código:

```
estado=xpath(MS_CambioEstadoSolicitudJefeInmediato,"string(//*[local-name()='strEstado'])");  
  
strcorreoempleado=xpath(MS_CambioEstadoSolicitudJefeInmediato,"string(//*[local-name()='strcorreoempleado'])");  
  
strcorreocontador=xpath(MS_CambioEstadoSolicitudJefeInmediato,"string(//*[local-name()='strcorreocontador'])");
```

Código 16. Extracción de variables

3. Se pregunta por el estado de la solicitud que fue seleccionada por el jefe inmediato, éstos pueden ser: **AprobadaXJefeInmediato** o **DesaprobadaXJefeInmediato**.
En caso de ser **AprobadaXJefeInmediato** el flujo sigue por el lado izquierdo el cual tiene la numeración empezada con 4, en caso contrario, es decir **DesaprobadaXJefeInmediato** el flujo sigue por el lado derecha el cual tiene la numeración empezada con 5.
4. Flujo de aprobación por jefe inmediato:
 - a. Se construye el mensaje Aprobar Vacaciones Jefe Inmediato Request.
 - b. Se llama al webmethod **AprobarVacacionesJefe**.
 - c. Se recibe la respuesta del webservice.
 - d. Se construye el mensaje **EnviarCorreoAprobacionEmpleado Request** mediante el cual se envía notificación al empleado que su solicitud ha sido aprobada.

- e. Se construye el mensaje **EnviarSparkAprobacionEmpleado Request** mediante el cual se envía notificación al empleado que su solicitud ha sido aprobada.
- f. Se llama al webmethod **EnviarCorreo** del webservice **WsNotificaciones** con el
- g. Se recibe la respuesta del webservice.
- h. Se llama al webmethod **EnviarSpark** del webservice **WsNotificaciones**.
- i. Se recibe la respuesta del webservice.
- j. Se construye el mensaje **EnviarCorreoContadorAprobacion Request**, mediante el cual se envía notificación al contador que una solicitud de un empleado ha sido aprobada.
- k. Se construye el mensaje **EnviarSparkContadorAprobacion Request**, mediante el cual se envía notificación al contador que una solicitud de un empleado ha sido aprobada.
- l. Se llama al webmethod **EnviarCorreo** del webservice **WsNotificaciones**.
- m. Se recibe la respuesta del webservice.
- n. Se llama al webmethod **EnviarSpark** del webservice **WsNotificaciones**.
- o. Se recibe la respuesta del webservice.

5. Flujo de desaprobación por jefe inmediato:

- a. Se construye el mensaje **DesaprobarVacacionesJefe Request**.
- b. Se llama al webmethod **DesaprobarVacacionesJefe** del servicio **WsVacaciones**.
- c. Se recibe la respuesta del servicio.
- d. Se construye el mensaje **EnviarCorreoEmpleado Request**.
- e. Se construye el mensaje **EnviarSparkEmpleado Request**.
- f. Se llama al webmethod **EnviarCorreo** del servicio **WsNotificaciones**.
- g. Se recibe la respuesta del servicio.
- h. Se llama al webmethod **EnviarSpark** del servicio **WsNotificaciones**.
- i. Se recibe la respuesta del servicio.

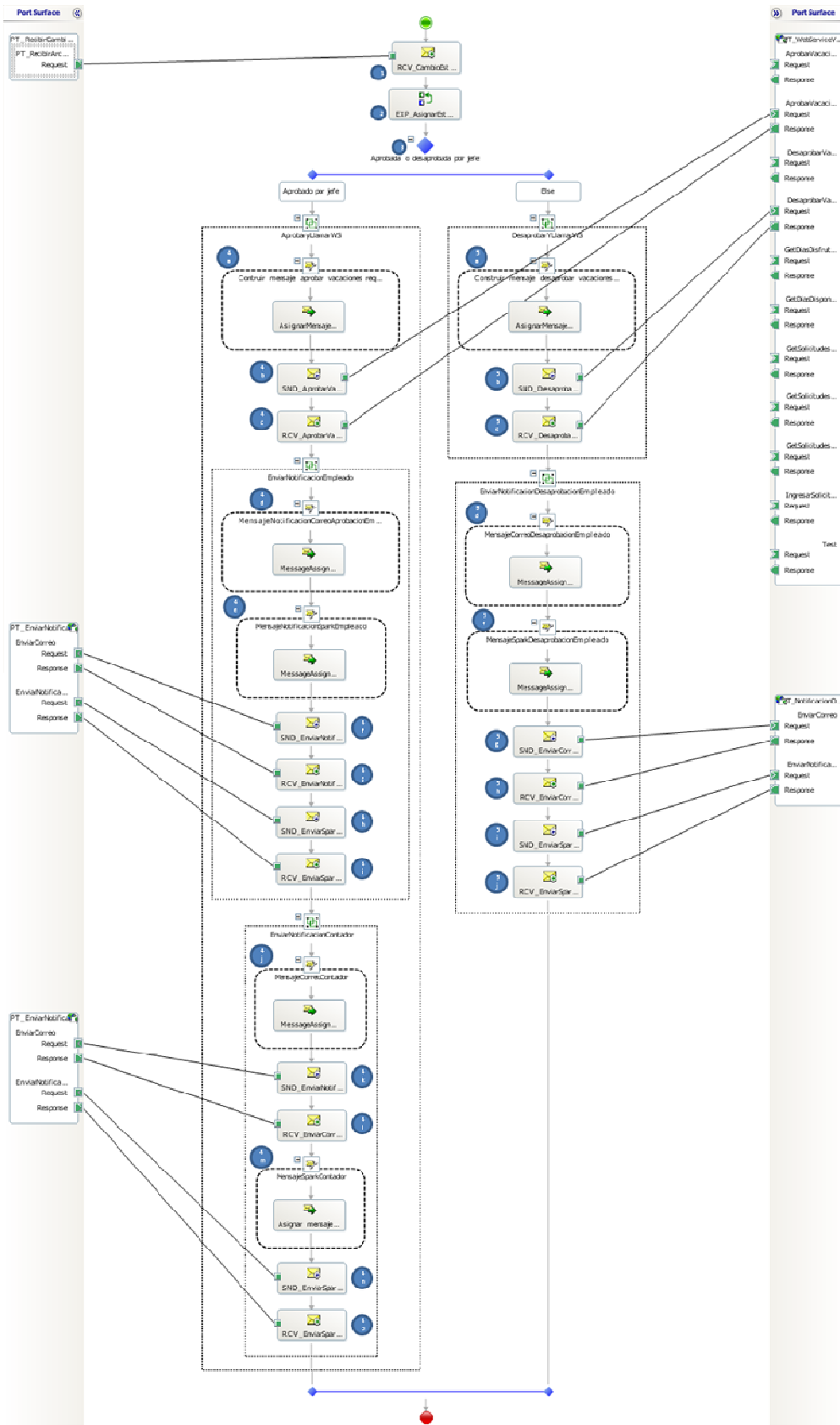


Figura 22. Flujo de trabajo de cambio de solicitud contador jefe en Biztalk

3. Cambio Estado Solicitud Contador:

Este flujo de trabajo, recibe un archivo XML en la carpeta configurada en el archivo web.config con la clave **folderCambioEstadoContador**.

```
<add key="folderCambioEstadoContador"  
value="C:\ProcesoVacaciones\CambiarEstadoContador\"/>
```

Código 17. FolderCambioEstadoContador en el web.config

Los archivos que llegan a esta carpeta deben tener el siguiente esquema:

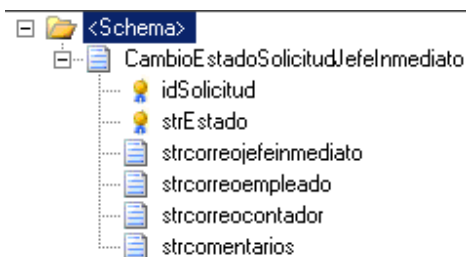


Figura 23. Esquema de archivos de cambio solicitud contador

```
<?XML version="1.0" ?>  
  
<ns0:CambioEstadoSolicitudJefeInmediato  
xmlns:ns0="http://BTS_ProcesoVacaciones2.CambioEstadoSolicitud">  
  
  <idSolicitud>37</idSolicitud>  
  
  <strEstado>AprobadaContador</strEstado>  
  
  <strcorreojefeinmediato>levalencia@avansoft.com</strcorreojefeinmediato>  
  
  <strcorreoempleado>dkepes@avansoft.com</strcorreoempleado>
```

```
<strcorreocontador>dkepes@avansoft.com</strcorreocontador>
<strcomentarios />
</ns0:CambioEstadoSolicitudJefeInmediato>
```

Código 18. Ejemplo del esquema de archivos de cambio solicitud contador

Una vez el archivo es creado en la carpeta destino, el flujo de Biztalk corre de la siguiente manera:

1. Se lee el archivo XML de la carpeta **folderCambioEstadoContador** en el archivo web.config.
2. Se extraen las variables que servirán durante el resto del flujo con el siguiente código:

```
estado=xpath(MS_CambioEstadoSolicitud,"string(//*[local-
name()='strEstado']");
strcorreoempleado=xpath(MS_CambioEstadoSolicitud,"string(//*[local-
name()='strcorreoempleado']");
strcorreojefe=xpath(MS_CambioEstadoSolicitud,"string(//*[local-
name()='strcorreojefeinmediato']");
```

Código 19. Extracción de variables del esquema de cambio de solicitud contador

- a. Se pregunta por el estado de la solicitud seleccionado por el contador, estos pueden ser: **AprobadaXContador** o **DesaprobadaXContador**.
En caso de ser **AprobadaXContador** el flujo sigue por el lado izquierdo el cual tiene la numeración empezada con 3, en caso contrario, es decir **DesaprobadaXContador** el flujo sigue por el lado derecho el cual tiene la numeración empezada con 4.
3. Flujo de aprobación por contador:
 - a. Se construye el mensaje **AprobacionContador** request.
 - b. Se llama al web method **AprobarVacacionesContador** del servicio **WsVacaciones**.
 - c. Se recibe la respuesta del web service.
 - d. Se construye el mensaje **EnviarCorreoAprobacionEmpleado** request.
 - e. Se llama al web method **EnviarCorreo** del servicio **WsNotificaciones**.

- f. Se recibe la respuesta del web service.
 - g. Se construye el mensaje **EnviarSparkAprobacionEmpleado** request.
 - h. Se llama al web method **EnviarSpark** del servicio **WsNotificaciones**.
 - i. Se recibe la respuesta del web service.
4. Flujo de desaprobación por contador:
- a. Se construye el mensaje **DesaprobacionContador** request.
 - b. Se llama el webmethod **DesaprobarVacacionesContador** del servicio **WsVacaciones**.
 - c. Se recibe la respuesta del web service.
 - d. Se construye el mensaje **EnviarCorreoEmpleadoDesaprobacion** request.
 - e. Se llama el web method **EnviarCorreo** del servicio **WsNotificaciones**.
 - f. Se recibe la respuesta del servicio.
 - g. Se construye el mensaje **EnviarSparkEmpleadoDesaprobacion** request.
 - h. Se llama al web method **EnviarSpark** del servicio **WsNotificaciones**.
 - i. Se recibe la respuesta del servicio.



Figura 24. Diagrama del flujo de trabajo de cambio de solicitud contador

MANUAL DE INSTALACION

Esta sección del documento pretende ser una guía gráfica y descriptiva para la instalación de la aplicación con Windows Workflow Foundation y BizTalk Server.

Prerrequisitos:

- Windows 2003 Server Standard Edition o Enterprise Edition.
- .NET Framework 3.0.
- BizTalk Server 2006 (cualquier versión).
- SQL Server 2005.

1. Instalación de la base de datos.

Para la creación de la base de datos debe tener permisos de administrador en SQL Server 2005, luego ejecutar el siguiente script.

```
CREATE DATABASE [DBVACACIONES] ON PRIMARY
( NAME = N'dbVacaciones', FILENAME = N'E:\Databases\dbVacaciones.mdf' ,
SIZE = 2240KB , MAXSIZE = UNLIMITED, FILEGROWTH = 1024KB )
LOG ON
( NAME = N'dbVacaciones_log', FILENAME =
N'E:\Databases\dbVacaciones_log.LDF' , SIZE = 560KB , MAXSIZE = 2048GB
, FILEGROWTH = 10%)
COLLATE Modern_Spanish_CI_AS
GO
```

Código 20. Crear base de datos

Luego de haber creado la base de datos se debe ejecutar el siguiente script para poner la sesión en contexto de la base de datos creada:

```
USE DBVACACIONES
```

Código 21. Usar base de datos

Y por ultimo ubicar en los medios de instalación la carpeta Scripts y ejecutar el archivo Script. Sql para la creación de todo el modelo entidad relación.

2. Instalación de la aplicación.

En los medios de instalación ubicar la carpeta Setup y luego el archivo Setup.Exe y ejecutarlo.

- a. En el primer paso sólo debe dar click en Next.

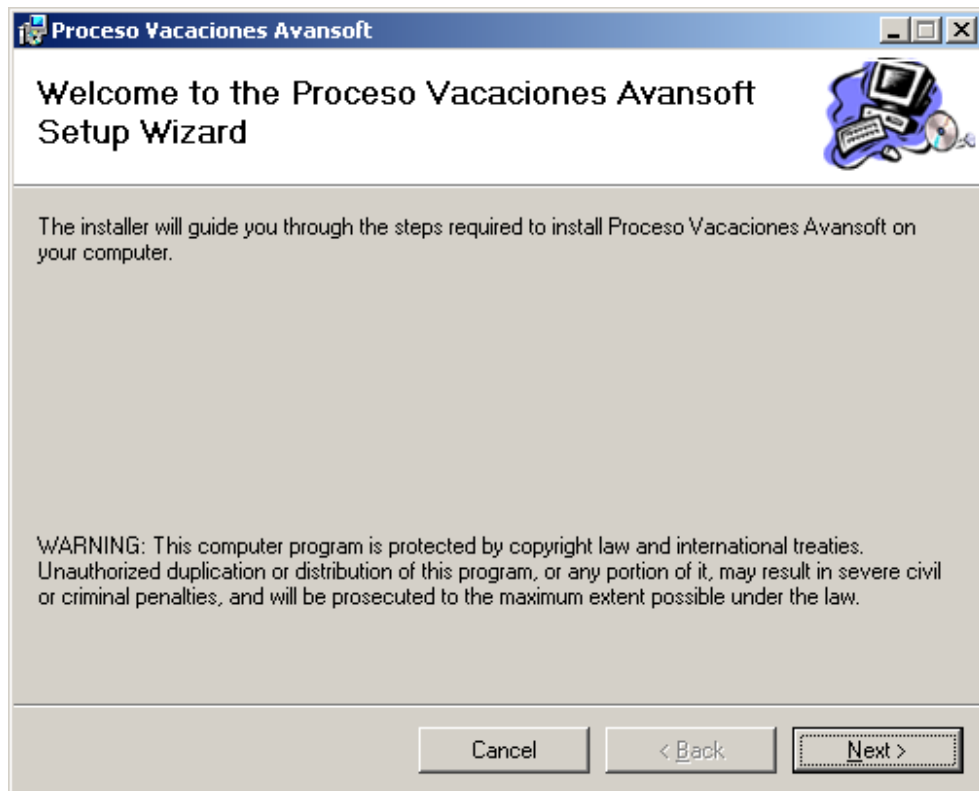


Figura 25. Instalación paso 1

- b. Luego el instalador pregunta el sitio web donde se desea instalar y el directorio virtual deseado.

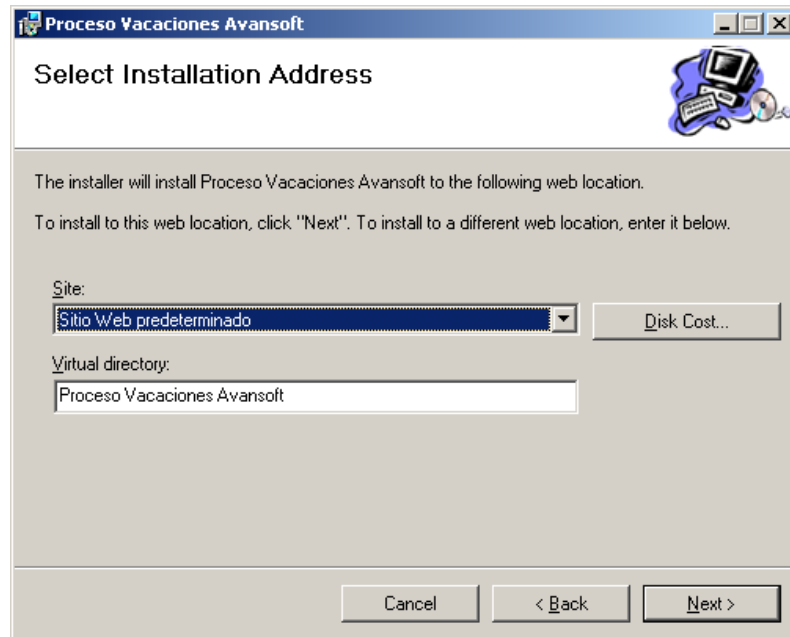


Figura 26. Instalación paso 2

- c. Luego debe dar click en siguiente y el instalador ejecuta todas sus acciones necesarias.

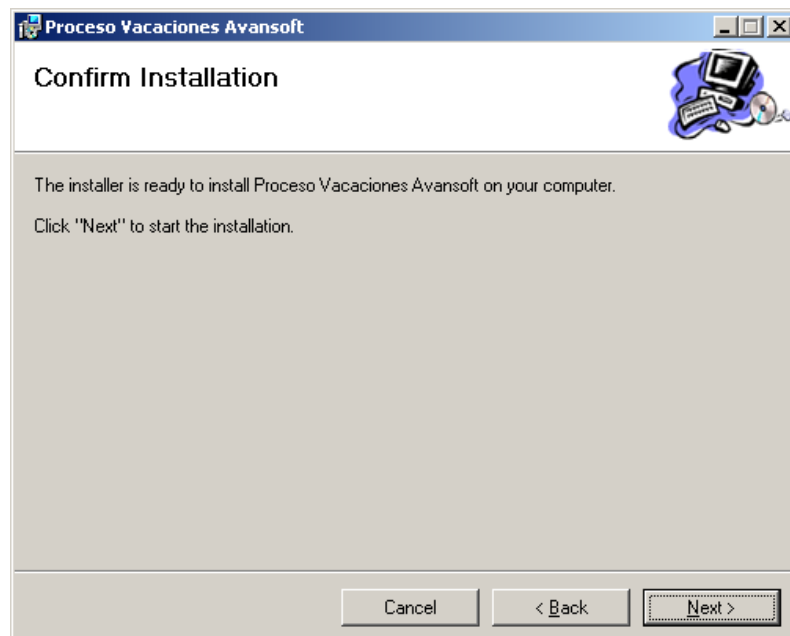


Figura 27. Instalación paso 3

3. Instalación de flujos de trabajo de BizTalk.

Para la correcta ejecución del flujo de trabajo se deben crear las siguientes carpetas en la unidad C:

1. ProcesoVacaciones
 1. CambioEstadoContador
 2. CambioEstadoJefe
 3. Entrada

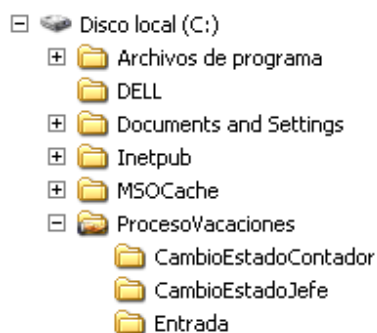


Figura 28. Instalación paso 4

Luego en la Carpeta Biztalk de los medios de instalación buscar el archivo Proceso Vacaciones Biztalk.msi y ejecutarlo.

1. Seleccionar la carpeta donde desea instalarlo y dar click en Next.

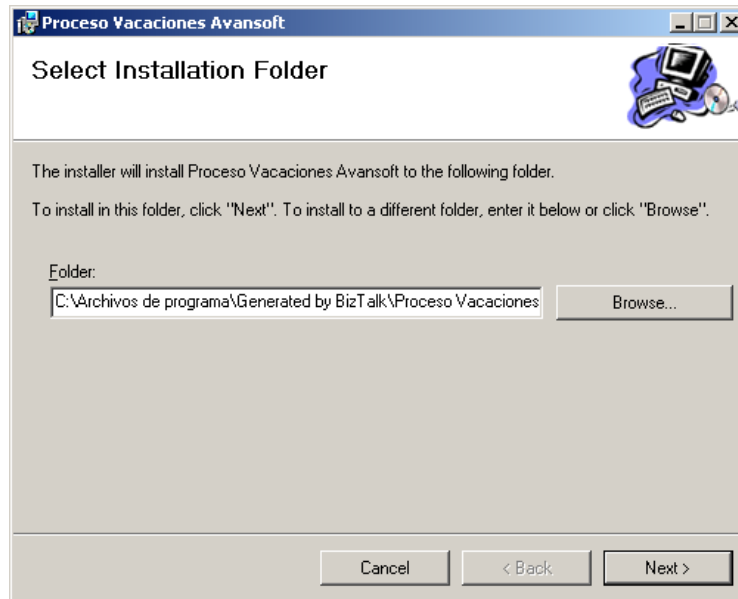


Figura 29. Instalación paso 5

2. En la pantalla de confirmación sólo es dar click en Next.

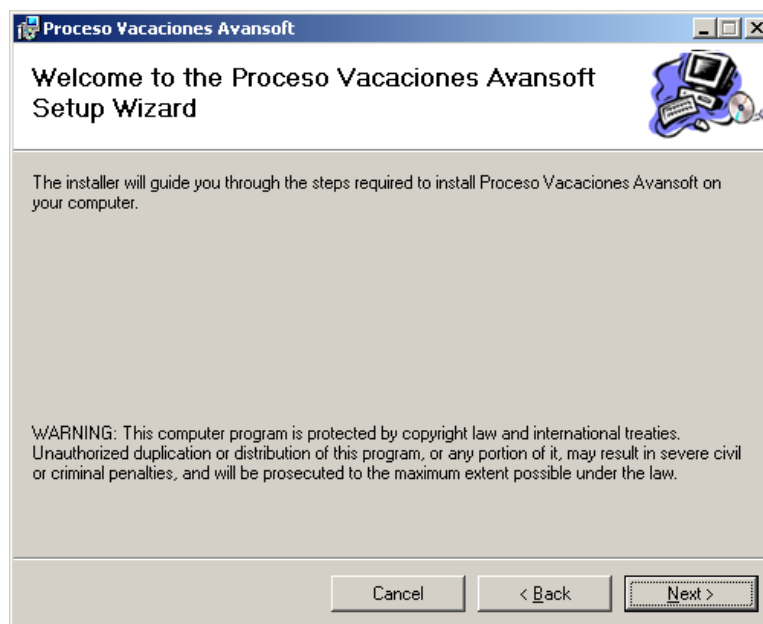


Figura 30. Instalación paso 6

3. La siguiente pantalla le pide al usuario que se debe importar el archivo MSI en la base de datos de administración de Biztalk Server, dar click en Next.

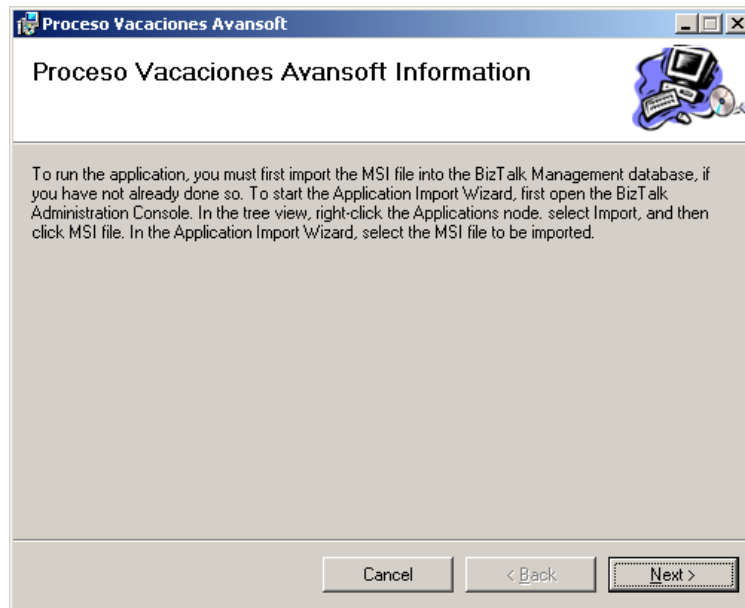


Figura 31. Instalación paso 7

4. Luego aparece la pantalla final de confirmación y sólo debe dar click en Close.

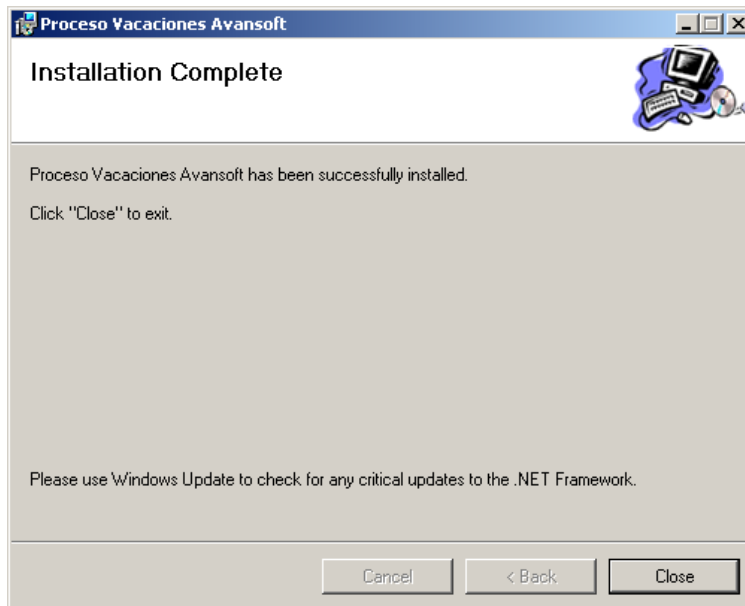


Figura 32. Instalación paso 8

5. Luego debe abrir el Biztalk Administration console, en Inicio -> Programas -> Biztalk Server 2006 -> Biztalk Server Administration.

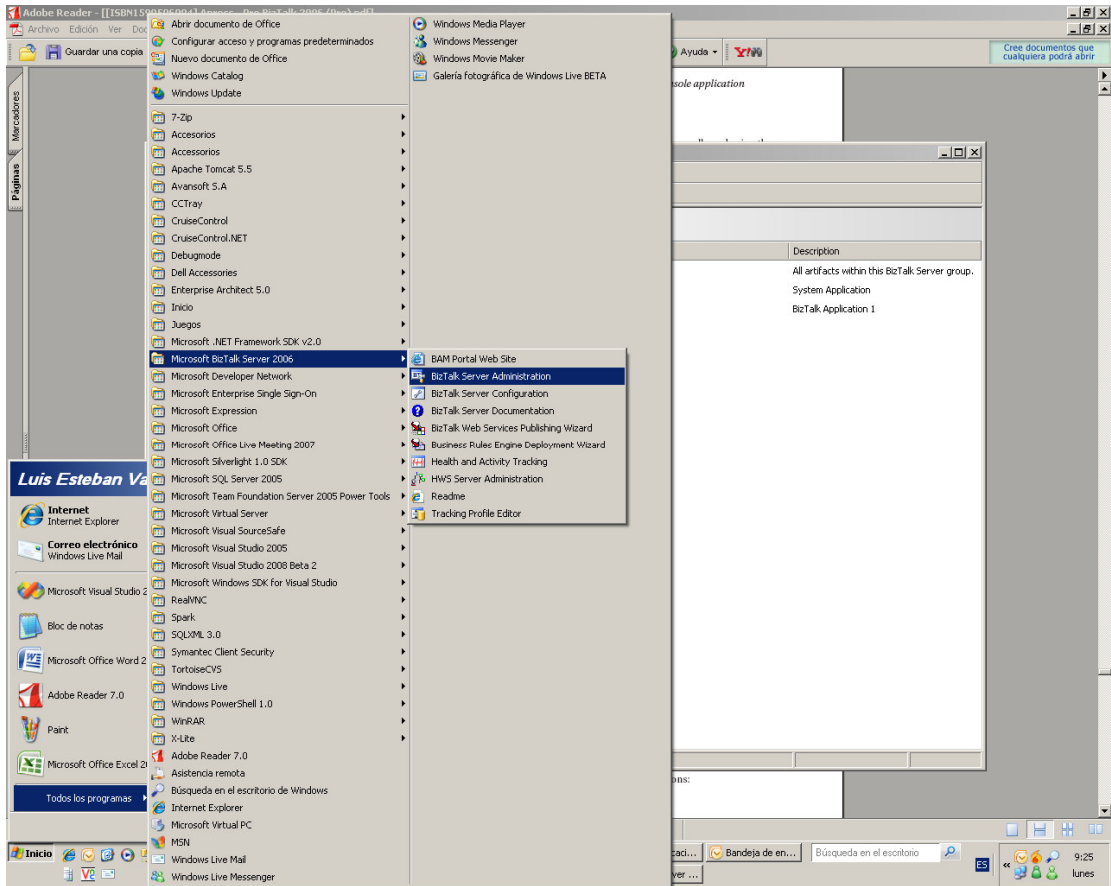


Figura 33. Instalación paso 9

6. Luego se debe importar el archivo MSI instalado anteriormente, para esto se abre el nodo Biztalk Server 2006 Administration, luego Biztalk Group, luego Application y en este con click derecho se selecciona la opción Import MSI FILE

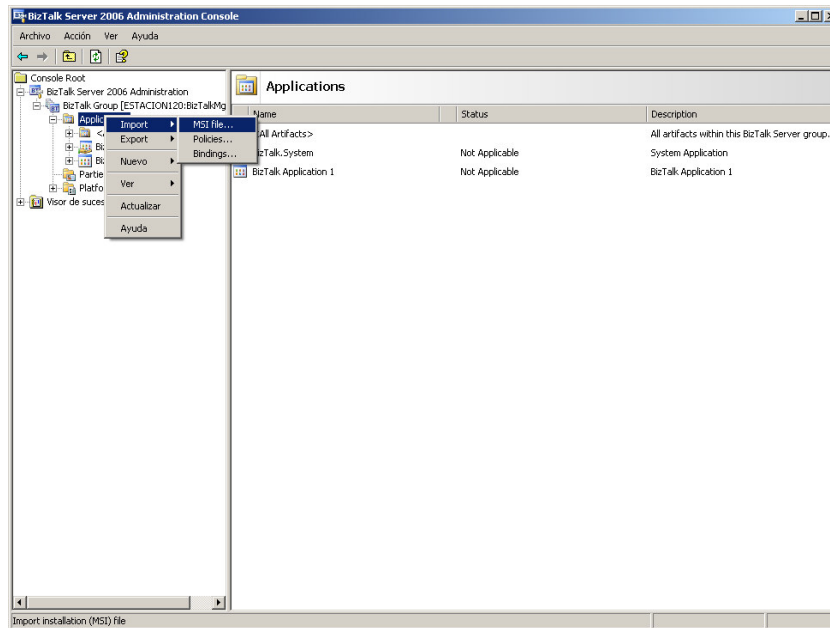


Figura 34. Instalación paso 10

7. En la pantalla siguiente se debe seleccionar el archivo MSI a importar.



Figura 35. Instalación paso 11

8. Luego en la pantalla Application Settings, seleccionar en la parte inferior Biztalk Application y Biztalk.System y dar click en Next.

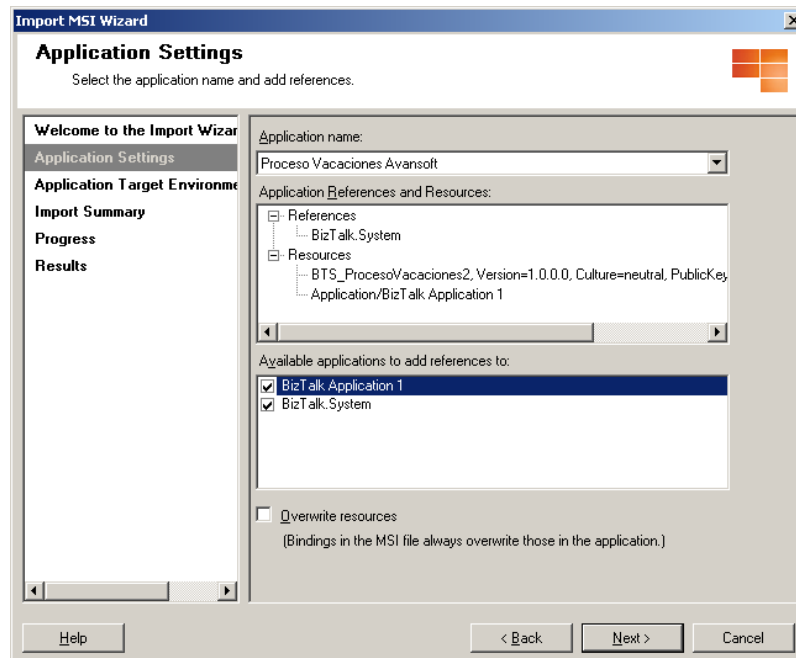


Figura 36. Instalación paso 12

9. Luego en la sección Application Target Environment, dejar seleccionado el Ambiente por defecto y dar click en Next:

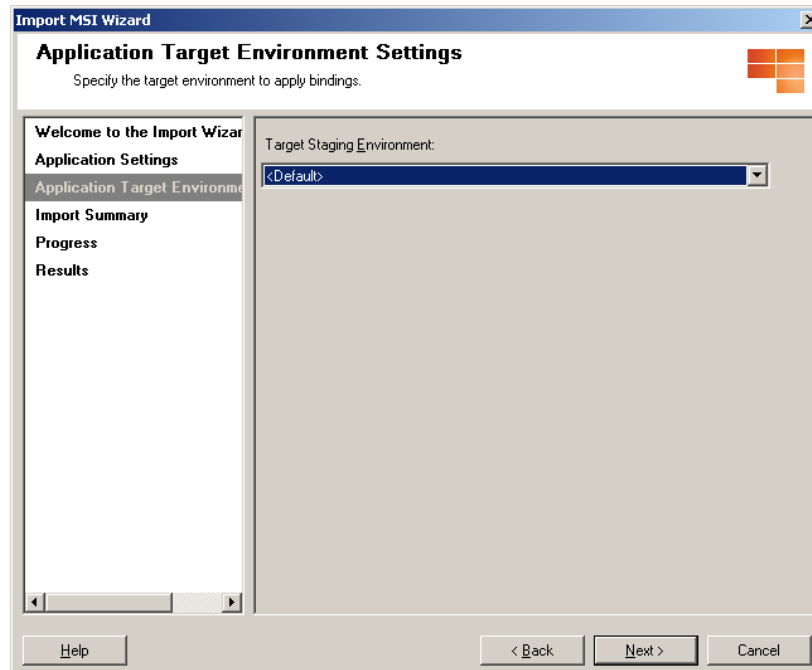


Figura 37. Instalación paso 13

10. Luego en la pantalla de Import Summary, simplemente darle click a Import.

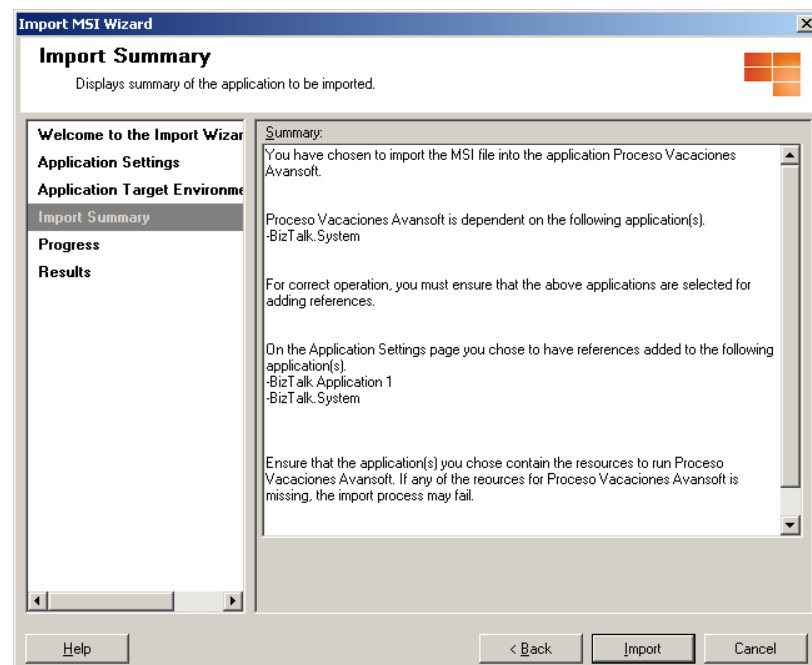


Figura 38. Instalación paso 14

11. Luego, esperar a que termine el proceso y que muestre la pantalla de resultados.

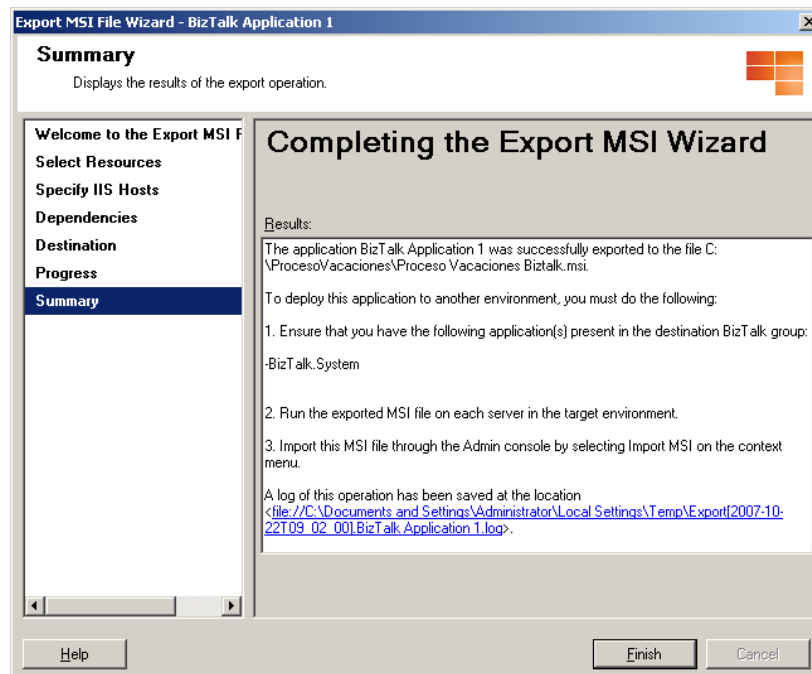


Figura 39. Instalación paso 15

12. Una vez terminada la instalación se procede a iniciar las orquestaciones y los puertos. En la consola de administración se abre el nodo Biztalk Server 2006 Administration, luego Biztalk Group, luego Application y por último se selecciona la aplicación Biztalk Application 1, luego se selecciona orquestaciones, para cada orquestación se hace lo siguiente:

- Click derecho Enlist.
- Click derecho Start.

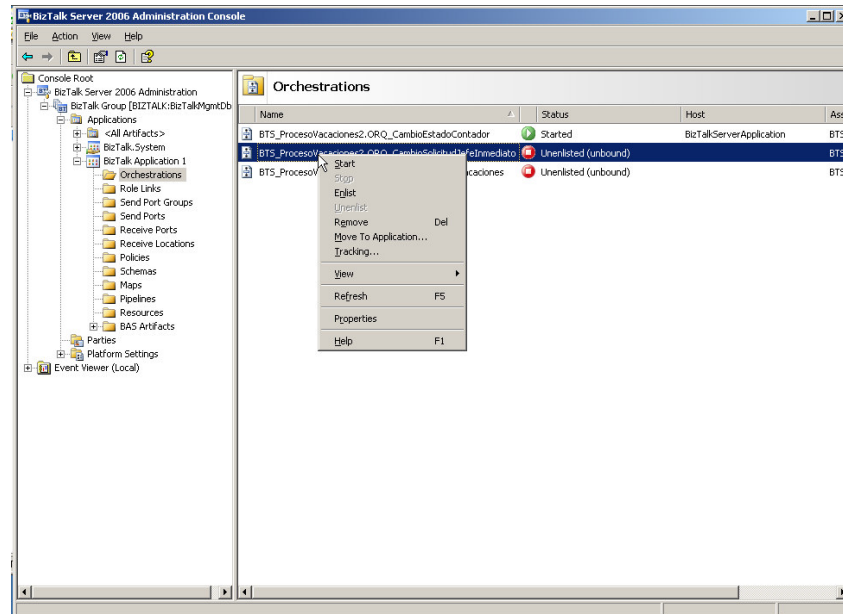


Figura 40. Instalación paso 16

13. Se selecciona en el menú de la izquierda Send Ports y por cada puerto existente se hace lo siguiente:

- a. Click derecho Enlist.
- b. Click derecho Start.

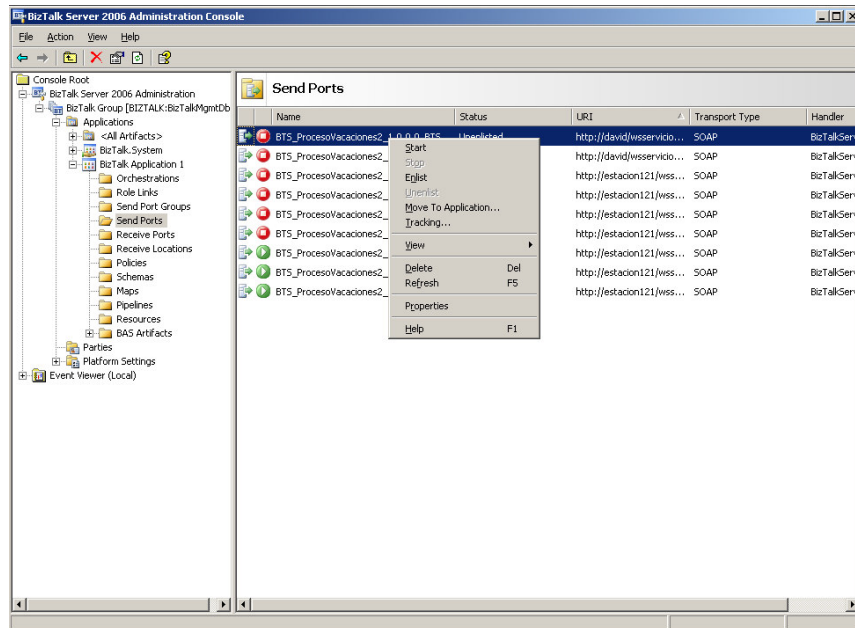


Figura 41. Instalación paso 17

14. Por último se selecciona en el menú de la izquierda Receive Locations y por cada ítem se realiza lo siguiente:

- a. Click derecho Enable.

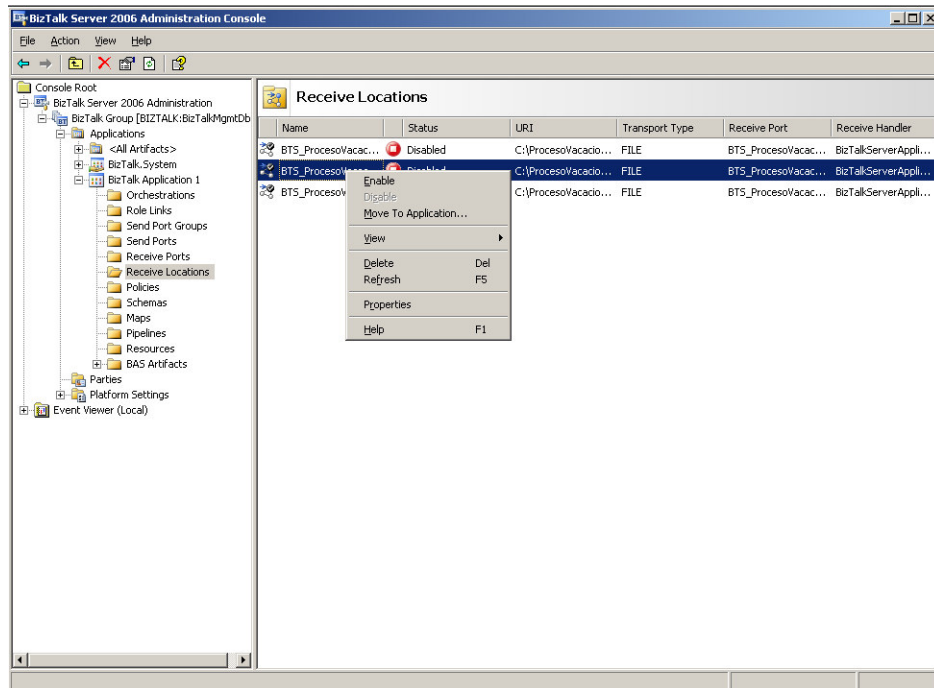


Figura 42. Instalación paso 18

Eso es todo para que la orquestación quede funcionando con la aplicación del proceso de vacaciones de Avansoft.

4. Desinstalación.

Para desinstalar el programa simplemente debe ingresar al panel de control, luego agregar o quitar programas, en la lista buscar Proceso Vacaciones Avansoft y dar click en quitar.

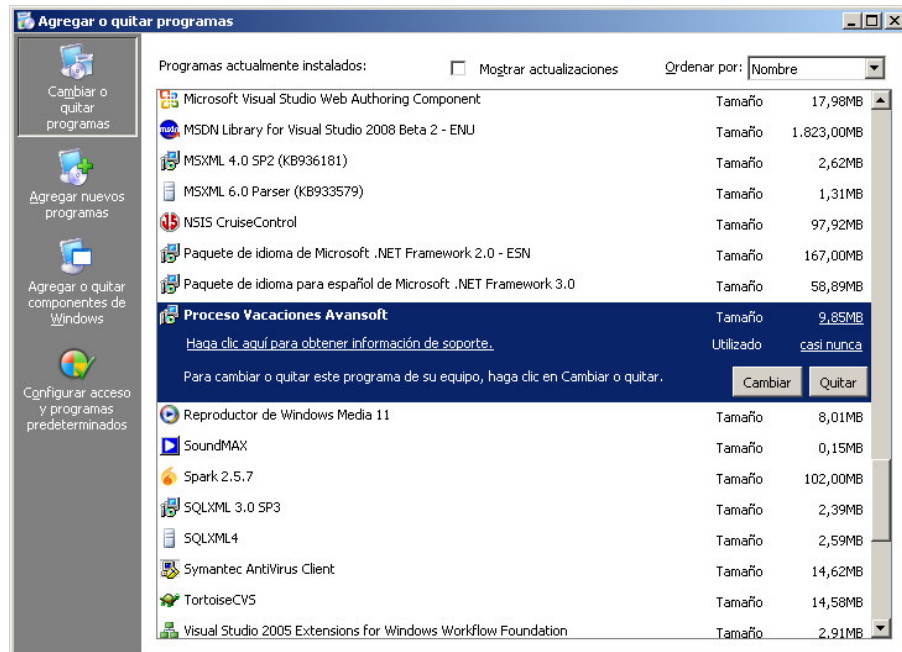


Figura 43. Desinstalación

TOMA DE DECISION

Para la toma de decisión decidimos utilizar los siguientes criterios:

- **Costo (5%):** Valor de la herramienta.
- **Facilidad de uso (5%):** Tiempo que toma el desarrollador en adaptarse a la tecnología
- **Curva de aprendizaje (5%):** Tiempo de aprendizaje de la herramienta para desarrollar procesos básicos.
- **Seguridad (10%):** Seguridad que implementa la herramienta para poder asegurar los procesos de negocio.
- **Estabilidad de la herramienta (10%):** Estabilidad de la herramienta y de sus componentes.
- **Facilidad de instalación (5%):** Es el tiempo necesario para aprender e instalar los componentes de cada herramienta.
- **Recursos requeridos (5%):** Recursos de hardware requeridos para poder utilizar las herramientas.

- **Mantenimiento (5%):** Es el tiempo necesario en administrar la herramienta y sus procesos.
- **Tracking (5%):** Es la posibilidad de hacer rastreo de los procesos de negocio, sus excepciones o estados erróneos y también sus procesos exitosos.
- **Backups de transporte (10%):** Es la posibilidad de que la herramienta utilice otro medio de envío de mensajes a los servicios en caso de que los principales no estén disponibles.
- **Productividad del desarrollador después del aprendizaje (10%):** Es el hecho de que la productividad del desarrollador aumente con respecto al uso de las herramientas tradicionales una vez su aprendizaje con estas herramientas haya sido exitoso.
- **Configuración (10%):** Posibilidad de la herramienta para ser configurable en tiempo de ejecución.
- **Robustez (10%):** Capacidad de la herramienta para procesar más usuarios o procesos con los mismos recursos.

TABLA DE TOMA DE DECISIONES

	Porcentaje	Biztalk Server	Calificación	WWF	Calificación
Costo.	5%	1	0,05	5	0,25
Facilidad de uso.	5%	4	0,2	4	0,2
Curva de aprendizaje.	5%	3	0,15	4	0,2
Seguridad	10%	5	0,5	5	0,5
Estabilidad de la herramienta	10%	5	0,5	5	0,5

Facilidad de instalación	5%	3	0,15	5	0,25
Recursos requeridos (Cpu, Memoria)	5%	3	0,15	5	0,25
Mantenimiento.	5%	5	0,25	3	0,15
Monitoreo.	5%	5	0,25	3	0,15
Tracking.	5%	5	0,25	3	0,15
Backups de transporte.	10%	5	0,5	1	0,1
Productividad del desarrollador después del aprendizaje.	10%	5	0,5	5	0,5
Configuración.	10%	5	0,5	5	0,5
Robustez.	10%	5	0,5	3	0,3
	100%		4,45		4

Tabla 7. Tabla de toma de decisiones de criterios

CONCLUSIONES DE LA TOMA DE DECISIÓN

- BiztalkServer es una herramienta más robusta y potente debido a que es una herramienta de Servidor mientras WF es simplemente un framework de desarrollo.

- BiztalkServer es la herramienta utilizada por Microsoft y todos sus socios de negocio para la implementación de procesos de negocio bajo una arquitectura orientada a servicios.
- Según la toma de decisiones es mejor utilizar BizTalk Server pero Windows WorkFlow Foundation permite realizar el mismo proceso de negocio sin necesidad de gastos.
- Windows Workflow Foundation no tiene las herramientas necesarias para crear un Enterprise Service Bus como lo tiene BizTalk Server.

CONCLUSIONES

- Después de una extensa investigación en ambas herramientas notamos que el aprendizaje de una nueva herramienta o frameworks implica casi la misma cantidad de tiempo, aunque éstos no tengan las mismas funcionalidades.
- La arquitectura orientada a servicios realmente beneficia a las empresas basadas en procesos, porque suponiendo que estos ya están desarrollados simplemente es hacer uso de ellos sin importar la lógica que hagan en su interior, siendo estos reutilizables y aumentando la productividad de los desarrolladores.
- Los servicios no se deben confundir con servicios web, existen servicios de forma binaria como DLL's y aunque estos no se alinean directamente con la arquitectura orientada a servicios, siguen siendo válidos por su reutilización e incremento de productividad.
- SOA es la estrategia que permite atacar esta complejidad y darnos un camino hacia la correcta gestión de la misma, donde radica el éxito de los sistemas. El reto actual de las empresas consiste no sólo "en mantener a raya" los costos que el mantenimiento supone, sino incluso, y a pesar de esta complejidad, su disminución. Hay que incorporar, por tanto, nuevas herramientas o estrategias para que esto sea posible, ya que hasta el momento muchos de los sistemas de gestión no se han administrado de forma directa sino a partir de la monitorización de su entorno de ejecución y detectando los comportamientos anómalos.
- Hablar de SOA no es hablar sólo de web services, sino que el tema va mucho más allá. El ciclo de vida de desarrollo de software, desde el diseño hasta la operación, está encima de la mesa para ser re-estudiado en base a las aportaciones y posibilidades de SOA. A medida que el mercado vaya adoptando las estrategias y tecnologías implicadas, veremos madurar el

desarrollo de software hasta niveles no alcanzados por el momento. El mundo del software no deja de reinventarse continuamente y es por esto que debemos estar preparados para adoptar las nuevas tecnologías, técnicas y metodologías alineadas siempre con las estrategias de negocio de la compañía.

- SOA es el estilo de arquitectura más importante del momento, en desarrollo simultáneo en la academia y la industria.
- Biztalk Server ofrece una plataforma robusta y ágil para el desarrollo de soluciones bajo el enfoque de arquitecturas orientadas a servicios.
- Windows Workflow Foundation ofrece una plataforma económica pero menos ágil para el desarrollo de soluciones bajo el enfoque de arquitecturas orientadas a servicios.
- Biztalk Server con su base de datos de administración ofrece la posibilidad de implementar un ESB empresarial, cosa que sólo se podrá implementar manualmente en Windows WorkFlow Foundation.

BIBLIOGRAFÍA

- [1] Surekha Durvasula. SOA Practitioner's guide Part1. Why Service Oriented Architecture? Septiembre2006.
- [2] Pro BizTalk 2006 - Appress - George Dunphy and Ahmed Metwally
- [3] BizTalk 2006 Recipes - A Problem-Solution Approach - Appress - Mark Beckner, Benjamin Goeltz, Brandon Gross, Brennan O'Reilly, Stephen Roger, Mark Smith, and Alexander West
- [4] Windows WorkFlow Foundation Step by Step - Wintellect - Kenn Scribner
- [5] Foundations of WF An Introduction to Windows Workflow Foundation - Appress - Brian R. Myers
- [6] Programming Windows Workflow Foundation Practical WF Techniques and Examples using XAML and C# - PACKT PUBLISHING - K. Scott