DESARROLLO DE UN PROTOTIPO DEL TUTOR KCMM QUE PERMITE SU ACTUALIZACIÓN POR MEDIO DE OBJETOS Y AGENTES MÓVILES

ANEXOS

ADRIANA MARITZA CORTÉS GARCÍA WILLIAN KLINGER MOSQUERA

UNIVERSIDAD DISTRITAL "FRANCISCO JOSÉ DE CALDAS"

FACULTAD DE INGENIERÍA

PROYECTO CURRICULAR INGENIERÍA DE SISTEMAS

BOGOTÁ, JUNIO 2003

TABLA DE CONTENIDO

1	ANE	KO A DI	AGRAMAS DE CASOS DE USO Y DIAGRAMAS DE SECUENCIA	21
ΟВ	SERV	ACIONE	:s	21
1.1	DIAG	RAMAS	DE CASOS DE USOS Y DIAGRAMAS DE SECUENCIAS KCAGENTES	
	(HERI	RAMIENTA	A PARA LA A CTUALIZACIÓN DE SOFTWARE)	22
	1.1.1	Diagran	na Gestión Versiones	22
		_	Adicionar Componente	
		1.1.1.2	Modificar Componente	26
		1.1.1.3	Eliminar Componente	28
		1.1.1.4	Adicionar Funcionalidad	30
		1.1.1.5	Modificar Funcionalidad	32
		1.1.1.6	Eliminar Funcionalidad	34
		1.1.1.7	Adicionar Versión	36
		1.1.1.8	Modificar Versión	38
		1.1.1.9	Eliminar Versión	40
		1.1.1.10	Planear Actualización	42
		1.1.1.11	Ejecutar Actualización	43
		1.1.1.12	Actualizar Versión	44
		1.1.1.13	Gestionar Base de Datos	46
	1.1.2	Diagran	na Gestión Usuarios	47
		1.1.2.1	Registrar Usuario	48
		1.1.2.2	Validar Usuario	50
		1.1.2.3	Habilitar Usuario	52
		1.1.2.4	Modificar Usuario	54
1.2	DIAG	RAMAS	DE CASOS DE USOS Y DIAGRAMAS DE SECUENCIAS TUTOR KCMM	
	VERS	SIÓN 2.0)	56
	1.2.1	Diagrar	na Gestión Información	56
		1.2.1.1	Consultar Información	57
		1.2.1.2	Evaluar Estudiante	59
2	ANE	OB D	ICCIONARIO DE CLASES	61
1.1	IOBJE	тоАсти	ALIZADOR	63
	1.1.1	Atributo	s	63
	1.1.2	Operac	iones	63
		1.1.2.1	Operación: ejecutarActualizacion	63
		1.1.2.2	Operación: aacAdicionarActividadActualizacion	63

		1.1.2.3	Operación: aacAdicionarActividadActualizacion	63
		1.1.2.4	Operación: aacEliminarActividad	63
		1.1.2.5	Operación: aacDevolverActividadActualizacion	63
		1.1.2.6	Operación: bEstaDisponible	63
1.2	STATI	ONARY A	GENT	64
	1.2.1	ATRIBU	JTOS	64
	1.2.2	OPERA	CIONES	64
1.3	PERS	ISTENTS	TATIONARY AGENT	65
	1.3.1	ATRIBU	JTOS	65
	1.3.2	OPERA	CIONES	65
1.4	ACTIV	/IDADACT	UALIZACION	66
	1.4.1	ATRIBU	JTOS	66
		1.4.1.1	Atributo: sld	66
		1.4.1.2	Atributo: ghaAgencia	66
		1.4.1.3	Atributo: dateFechaEjecucion	66
		1.4.1.4	Atributo: verVersion	66
		1.4.1.5	Atributo: sResponsable	66
		1.4.1.6	Atributo: bRealizada	66
	1.4.2	OPERA	CIONES	67
		1.4.2.1	Operación: ActividadActualizacion	67
1.5	IOBJE	TOCONS	ULTA	68
	1.5.1	ATRIBU	JTOS	68
	1.5.2	OPERA	CIONES	68
		1.5.2.1	Operación: consultar	68
		1.5.2.2	Operación: vtDevolverResultados	68
		1.5.2.3	Operación: bLeyoBaseDatos	68
		1.5.2.4	Operación: bConsultaFueRealizada	68
1.6	PERS	ISTENTM	OBILEA GENT	69
	1.6.1	6.1 ATRIBUTOS		
	1.6.2	OPERA	CIONES	69
1.7	FUNC	IONALIDA	ND	70
	1.7.1	ATRIBU	JTOS	70
		1.7.1.1	Atributo: sNombre	70
		1.7.1.2	Atributo: sDescripcion	70
		1.7.1.3	Atributo: htComponentes	70
	1.7.2	OPERA	CIONES	70
		1.7.2.1	Operación: Funcionalidad	70
		1722	Operación: comAdicionarComponente	70

		1.7.2.3	Operación: setDatos	.71
		1.7.2.4	Operación: getDatos	.71
		1.7.2.5	Operación: comEliminarComponente	.71
		1.7.2.6	Operación: comDevolverComponente	.71
		1.7.2.7	Operación: vtDevolverTodosComponentes	.71
		1.7.2.8	Operación: bEstaComponente	.71
1.8	СОМР	ONENTE		.72
	1.8.1	ATRIBL	JTOS	.72
		1.8.1.1	Atributo: iTipo	.72
		1.8.1.2	Atributo: sNombre	.72
		1.8.1.3	Atributo: sDesarrollador	.72
		1.8.1.4	Atributo: sDescripcion	.72
		1.8.1.5	Atributo: dateFechaCreacion	.72
		1.8.1.6	Atributo: sRutaClassName	.73
		1.8.1.7	Atributo: sRutaJava	.73
		1.8.1.8	Atributo: sRutaCodebaseUrl	.73
		1.8.1.9	Atributo: INTERFAZ_GRAFICA	.73
		1.8.1.10	Atributo: LOGICA_APLICACION	.73
	1.8.2	OPERA	CIONES	.73
		1.8.2.1	Operación: Componente	.73
		1.8.2.2	Operación: setDatos	.73
		1.8.2.3	Operación: bEsInterfazGrafica	.73
		1.8.2.4	Operación: getDatos	.74
1.9	OBJET	TOACTUA	LIZADOR	.75
	1.9.1	ATRIBL	JTOS	.75
		1.9.1.1	Atributo: vtActividadesActualizacion	.75
		1.9.1.2	Atributo: iPosActividadActual	.75
		1.9.1.3	Atributo: iEstadoActualizacion	.75
	1.9.2	OPERA	CIONES	.75
		1.9.2.1	Operación: ejecutarActividadActualizacion	.75
1 1() Vers	SION		76
			UTOS	
	1.10.1		Atributo: sNombre	
			Atributo: sNombre	
			Atributo: dateFechaCreacion	
			Atributo: dateFechaModificado	
			Atributo: sResponsable	
			Atributo: dTamano	
			Atributo: vtAsesores	
		1.10.1./	Allibulo. VIA3E301E3	. / /

1.10.1.8 Atributo: vtEstudiantes	77
1.10.1.9 Atributo: htFuncionalidades	77
1.10.2 OPERACIONES	77
1.10.2.1 Operación: Version	77
1.10.2.2 Operación: setDatos	77
1.10.2.3 Operación: getDatos	77
1.10.2.4 Operación: eliminarComponente	77
1.10.2.5 Operación: funAdicionarFuncionalidad	77
1.10.2.6 Operación: funDevolverFuncionalidad	77
1.10.2.7 Operación: funEliminarFuncionalidad	78
1.10.2.8 Operación: vtDevolverTodosComponentes	78
1.10.2.9 Operación: vtDevolverTodasFuncionalidades	78
1.10.2.1 Operación: vtDevolverTodosEstudiantes	78
1.10.2.11 Operación: vtDevolverAsesores	78
1.10.2.12 Operación: sBuscarEstudiante	78
1.11 ObjetoConsulta	79
1.11.1 ATRIBUTOS	79
1.11.1.1 Atributo: aiConsultor	79
1.11.1.2 Atributo: bConsultaRealizada	79
1.11.1.3 Atributo: blnvocoLeerBaseDatos	79
1.11.1.4 Atributo: proxyGestorPersistencia	79
1.11.1.5 Atributo: sConsultaSQL	80
1.11.1.6 Atributo: vtResultados	80
1.11.2 OPERACIONES	80
1.11.2.1 Operación: bCargarServidorGestorPersistencia	80
1.11.2.2 Operación: leerBaseDatos	80
1.11.2.3 Operación: viajarAdministrador	80
1.12 IREGION	81
1.12.1 ATRIBUTOS	81
1.12.2 OPERACIONES	81
1.13 GESTORV ERSIONES	
1.13.1 ATRIBUTOS	
1.13.1.1 Atributo: htVersiones	
1.13.1.2 Atributo: htEstudiantes	
1.13.1.3 Atributo: proxyObjetoActualizador	
1.13.1.4 Atributo: infoAgentes	
1.13.2 OPERACIONES	
1.13.2.1 Operación: aiCargarServidorObjeto	
1.13.2.2 Operación: bCargarServidorObjetoActualizador	82

1.14 IGESTORV ERSIONES	83
1.14.1 ATRIBUTOS	83
1.14.2 OPERACIONES	83
1.14.2.1 Operación: verAdicionarVersion	83
1.14.2.2 Operación: vtDevolverTodasVersiones	83
1.14.2.3 Operación: verDevolverVersion	83
1.14.2.4 Operación: verEliminarVersion	83
1.14.2.5 Operación: bEstaVersion	83
1.14.2.6 Operación: funAdicionarFuncionalidad	83
1.14.2.7 Operación: funEliminarFuncionalidad	84
1.14.2.8 Operación: comAdicionarComponente	84
1.14.2.9 Operación: comEliminarComponente	84
1.14.2.10 Operación: vtDevolverTodosEstudiantesVersion	84
1.14.2.11 Operación: adicionarEstudianteVersion	
1.14.2.12 Operación: adicionarTodosEstudiantesVersion	
1.14.2.13 Operación: instalarVersion	
1.15 ADMINISTRADOR	86
1.16 ATRIBUTOS	86
1.17 OPERACIONES	86
1.17.1.1 Operación: Administrador	86
1.17.1.2 Operación: sOperaciones	86
1.17.1.3 Operación: asignarOperacion	86
1.17.1.4 Operación: sDevolverOperacion	86
1.18 PERSISTENTS TATIONARY AGENT	87
1.19 ATRIBUTOS	87
1.20 OPERACIONES	87
1.21 IGESTORVERSIONES	88
1.22 ATRIBUTOS	88
1.23 OPERACIONES	88
1.23.1.1 Operación: verAdicionarVersion	
1.23.1.2 Operación: vtDevolverTodasVersiones	
1.23.1.3 Operación: verDevolverVersion	
1.23.1.4 Operación: verEliminarVersion	
1.23.1.5 Operación: bEstaVersion	
1.23.1.6 Operación: funAdicionarFuncionalidad	
1.23.1.7 Operación: funEliminarFuncionalidad	
1.23.1.8 Operación: comAdicionarComponente	
1.23.1.9 Operación: comEliminarComponente	
1.23.1.10 Operación: vtDevolverTodosEstudiantesVersion	

1.23.1.11 Operación: adicionarEstudianteVersion	89
1.23.1.12 Operación: adicionarTodosEstudiantesVersion	89
1.23.1.13 Operación: instalarVersion	89
1.24 ESTUDIANTE	90
1.25 ATRIBUTOS	90
1.25.1.1 Atributo: iNivelActual	90
1.25.1.2 Atributo: sPerfil	90
1.25.1.3 Atributo: vtCuestionario	90
1.25.1.4 Atributo: vtRespuestas	90
1.26 OPERACIONES	91
1.26.1.1 Operación: asignarCuestionario	91
1.26.1.2 Operación: asignarRespuestas	91
1.26.1.3 Operación: Estudiante	91
1.26.1.4 Operación: Estudiante	91
1.26.1.5 Operación: vtDevolverCuestionario	91
1.26.1.6 Operación: vtDevolverRespuestas	91
1.27 IGESTORUSUARIOS	92
1.27.1 ATRIBUTOS	92
1.27.2 OPERACIONES	92
1.27.2.1 Operación: admAdicionarAdministrador	92
1.27.2.2 Operación: admDevolverAdministrador	92
1.27.2.3 Operación: admDevolverAdministradorActual	92
1.27.2.4 Operación: asignarNivelEstudiante	92
1.27.2.5 Operación: crearEstudiante	92
1.27.2.6 Operación: estDevolverEstudiante	92
1.27.2.7 Operación: estAdicionarEstudiante	92
1.27.2.8 Operación: htDevolverTodosEstudiantes	93
1.27.2.9 Operación: editarAdministrador	93
1.27.2.10 Operación: editarEstudiante	
1.27.2.11 Operación: estDevolverEstudiante	
1.27.2.12 Operación: bGuardarEstudianteBaseDatos	
1.27.2.13 Operación: iDevolverEstadoLectura	
1.27.2.14 Operación: vtComprobarDatos	
1.27.2.15 Operación: vtDevolverTodosAdministradores	
1.28 GESTORUSUARIOS	
1.28.1 ATRIBUTOS	94
1.28.1.1 Atributo: admAdministrador	94
1.28.1.2 Atributo: estEstudiante	94
1.28.1.3 Atributo: htAdministrador	94

1.28.1.4 Atributo: iEstadoLectura	94
1.28.1.5 Atributo: proxyGestorVersiones	94
1.28.1.6 Atributo: proxyObjetoConsulta	94
1.28.2 OPERACIONES	95
1.28.2.1 Operación: bCargarServidorGestorVersiones	95
1.28.2.2 Operación: bCargarServidorObjetoConsulta	95
1.29 GESTORV ERSIONES	96
1.29.1 ATRIBUTOS	96
1.29.1.1 Atributo: htVersiones	96
1.29.1.2 Atributo: htEstudiantes	96
1.29.1.3 Atributo: proxyObjetoActualizador	96
1.29.1.4 Atributo: infoAgentes	96
1.29.2 OPERACIONES	96
1.29.2.1 Operación: aiCargarServidorObjeto	96
1.29.2.2 Operación: bCargarServidorObjetoActualizador	96
1.30 USUARIO	97
1.30.1 ATRIBUTOS	97
1.30.1.1 Atributo: sContrasena	97
1.30.1.2 Atributo: sCorreoElectronico	97
1.30.1.3 Atributo: sldentificacion	97
1.30.1.4 Atributo: sNombre	97
1.30.1.5 Atributo: sPrimerApellido	97
1.30.1.6 Atributo: sSegundoApellido	97
1.30.2 OPERACIONES	98
1.30.2.1 Operación: bComprobarClave	98
1.30.2.2 Operación: setDatos	98
1.30.2.3 Operación: getDatos	98
1.31 CONCEPTO	100
1.31.1 ATRIBUTOS	100
1.31.1.1 Atributo: ild	100
1.31.1.2 Atributo: iNivel	100
1.31.1.3 Atributo: sNombre	100
1.31.1.4 Atributo: sDefinicion	100
1.31.2 OPERACIONES	100
1.31.2.1 Operación: Concepto	100
1.31.2.2 Operación: setDatos	100
1.31.2.3 Operación: getDatos	101
1.32 CARACTERISTICA COMUN	102
1 32 1 ATRIBUTOS	102

1.32.2 OPERACIONES	102
1.32.2.1 Operación: CaracteristicaComun	102
1.33 PERSISTENTMOBILEAGENT	103
1.33.1 ATRIBUTOS	103
1.33.2 OPERACIONES	103
1.34 PRACTICA CLAVE	104
1.34.1 ATRIBUTOS	104
1.34.1.1 Atributo: sArea	104
1.34.1.2 Atributo: iCaraComun	104
1.34.2 OPERACIONES	104
1.34.2.1 Operación: PracticaClave	104
1.35 INIVEL	105
1.35.1 ATRIBUTOS	105
1.35.2 OPERACIONES	105
1.35.2.1 Operación: asignarld	105
1.35.2.2 Operación: cargarNivel	105
1.35.2.3 Operación: iDevolverEstadoLectura	105
1.35.2.4 Operación: iDevolverId	105
1.35.2.5 Operación: sDevolverBeneficios	105
1.35.2.6 Operación: sDevolverDeficion	105
1.35.2.7 Operación: sDevolverDefinicionArea	105
1.35.2.8 Operación: sDevolverDefinicionPractica	
1.35.2.9 Operación: sDevolverNombre	
1.35.2.10 Operación: sDevolverVisibilidad	
1.35.2.11 Operación: vtDevolverNombresAreasClaves	
1.35.2.12 Operación: vtDevolverNombresPracticas	
1.35.2.13 Operación: vtDevolverNombresPracticas	
1.36 AREACLAVE	
1.36.1 ATRIBUTOS	
1.36.1.1 Atributo: htPracticasClaves	
1.36.2 OPERACIONES	
1.36.2.1 Operación: AreaClave	
1.36.2.2 Operación: pcAsignarPracticaClave	
1.36.2.3 Operación: pcDevolverPracticaClave	
1.36.2.4 Operación: pcDevolverPracticaClave	
1.36.2.5 Operación: pcDevolverPracticasClaves	
1.36.2.7 Operación: vtDevolverNombresPracticas	107
LOUZE ANDERGOUL VIDEVOIVEHNUHURGERGUIGG	

1.37 SWCMM	108
1.37.1 ATRIBUTOS	108
1.37.1.1 Atributo: iEstadoLectura	108
1.37.1.2 Atributo: proxyObjetoConsulta	108
1.37.1.3 Atributo: sAreaClave	108
1.37.1.4 Atributo: sArquitectura	108
1.37.1.5 Atributo: sCaracteristicas	108
1.37.1.6 Atributo: sCaracteristicaComun	109
1.37.1.7 Atributo: sDefinicion	109
1.37.1.8 Atributo: sHistoria	109
1.37.1.9 Atributo: sNivel	109
1.37.1.10 Atributo: sNombre	109
1.37.1.11 Atributo: sPracticaClave	109
1.37.1.12 Atributo: vtCaracteristicasComunes	109
1.37.2 OPERACIONES	110
1.37.2.1 Operación: bCargarServidorObjetoConsulta	110
1.38 NIVEL	111
1.38.1 ATRIBUTOS	111
1.38.1.1 Atributo: iEstadoLectura	111
1.38.1.2 Atributo: ild	111
1.38.1.3 Atributo: htAreasClaves	111
1.38.1.4 Atributo: proxyObjetoConsulta	111
1.38.1.5 Atributo: sBeneficios	111
1.38.1.6 Atributo: sDefinicion	112
1.38.1.7 Atributo: sNombre	112
1.38.1.8 Atributo: sTransicion	112
1.38.1.9 Atributo: sVisibilidad	112
1.38.2 OPERACIONES	112
1.38.2.1 Operación: apcAdicionarAreaClave	112
1.38.2.2 Operación: apcDevolverAreaClave	112
1.38.2.3 Operación: apcEliminarAreaClave	112
1.39 ISWCMM	113
1.39.1 ATRIBUTOS	113
1.39.2 OPERACIONES	113
1.39.2.1 Operación: cargarSwCmm	
1.39.2.2 Operación: ccDevolverCaracteristicaComun	
1.39.2.3 Operación: iDevolverEstadoLectura	
1.39.2.4 Operación: sDevolverAreaClave	
1.39.2.5 Operación: sDevolverArquitectura	

1.39.2.6 Operación: sDevolverCaracteristicaComun	113
1.39.2.7 Operación: sDevolverCaracteristicas	113
1.39.2.8 Operación: sDevolverHistoria	113
1.39.2.9 Operación: sDevolverNivel	114
1.39.2.10 Operación: sDevolverNombre	114
1.39.2.11 Operación: vtDevolverTodasCaracteristicasComunes	114
1.40 PREGUNTA	116
1.40.1 ATRIBUTOS	116
1.40.1.1 Atributo: ild	116
1.40.1.2 Atributo: iLogro	116
1.40.1.3 Atributo: iNivel	116
1.40.1.4 Atributo: sEnunciado	116
1.40.1.5 Atributo: sOpcionA	116
1.40.1.6 Atributo: sOpcionB	117
1.40.1.7 Atributo: sOpcionC	117
1.40.1.8 Atributo: sOpcionD	117
1.40.1.9 Atributo: sOpcionE	117
1.40.1.10 Atributo: sRespuesta	117
1.40.2 OPERACIONES	117
1.40.2.1 Operación: Pregunta	117
1.41 LOGRO	118
1.41.1 ATRIBUTOS	118
1.41.1.1 Atributo: iDificultad	118
1.41.1.2 Atributo: ild	118
1.41.1.3 Atributo: iPreguntaActual	118
1.41.1.4 Atributo: prgPreguntas	118
1.41.1.5 Atributo: sEnunciado	118
1.41.1.6 Atributo: sTemaEstudiar	119
1.41.2 OPERACIONES	119
1.41.2.1 Operación: AsignarPregunta	119
1.41.2.2 Operación: bAdicionarPregunta	119
1.41.2.3 Operación: Logro	119
1.41.2.4 Operación: prgDevolverPregunta	119
1.41.2.5 Operación: prgDevolverPregunta	119
1.42 CUESTIONARIO	120
1.42.1 ATRIBUTOS	120
1.42.1.1 Atributo: IgrLogros	120
1.42.2 OPERACIONES	
1 42 2 1 Operación: asignarl ogro	120

	1.42.2.	2 Operación: asignarLogros	120
	1.42.2.	3 Operación: Cuestionario	120
	1.42.2.	4 Operación: IgrDevolverLogro	120
		5 Operación: prgDevolverPregunta	
1.43	B EVALUACION	N	121
	1.43.1 ATRI	BUTOS	121
	1.43.1.	1 Atributo: cueCuestionario	121
	1.43.1.	2 Atributo: iEstadoLectura	121
	1.43.1.	3 Atributo: proxyObjetoConsulta	121
	1.43.1.	4 Atributo: sRespuestas	121
	1.43.2 OPE	RACIONES	121
1.44	IEVALUACIO	N	122
	1.44.1 ATRI	BUTOS	122
	1.44.2 OPE	RACIONES	122
	1.44.2.	1 Operación: asignarCuestionario	122
		2 Operación: asignarRespuesta	
	1.44.2.	3 Operación: asignarRespuestas	122
	1.44.2.	4 Operación: cargarCuestionario	122
	1.44.2.	5 Operación: cueDevolverCuestionario	122
	1.44.2.	6 Operación: iCalificarCuantitativo	122
	1.44.2.	7 Operación: iDevolverEstadoLectura	122
	1.44.2.	8 Operación: prgDevolverPregunta	123
	1.44.2.	9 Operación: sCalificarCualitativo	123
	1.44.2.	.10 Operación: sDevolverRespuesta	123
	1.44.2.	.11 Operación: sDevolverRespuestas	123
1.45	PERSISTENT	MobileAgent	124
	1.45.1 ATRI	BUTOS	124
	1.45.2 OPE	RACIONES	124
3	ANEXO C N	IANUAL DE CONCEPTOS BÁSICOS KCAGENTES	125
3.1	PREFACIO.		125
	3.1.1 Acerca	a de este Documento	125
		nentos Relacionados	
		DS	
		nimiento o Actualización de Software	
	3.2.1.1	Clasificación de las Actualizaciones	
		3.2.1.1.1 Segun su Causa	128
		5.Z. L. L.Z. Seguil el Recuiso	178

			3.2.1.1.3 Según Su Naturaleza	129
	3.2.2	Tecnolo	gía de Agentes Móviles	.129
		3.2.2.1	Agente móvil	129
		3.2.2.2	Ambiente de agentes distribuidos (Ver Figura 4)	131
		3.2.2.3	Lugares	131
		3.2.2.4	Agencias	131
		3.2.2.5	Región	134
	3.2.3	Ambien	te para Mantenimiento de Software	.134
		3.2.3.1	Componente	134
		3.2.3.2	Funcionalidad	135
		3.2.3.3	Versión	135
		3.2.3.4	Proyecto KCAgentes	136
4	ANEX	O D MA	ANUAL DEL USUARIO KCAGENTES	.137
4.1	PREF	ACIO		. 137
	4.1.1	Acerca	de este Documento	. 137
	4.1.2	Docume	entos Relacionados	.137
4.2	INST	ALACIÓI	N	.139
	4.2.1	Requeri	mientos	.139
		•	o de Instalación	
		4.2.2.1	Preparando la Instalación	139
		4.2.2.2	Durante la Instalación	142
		4.2.2.3	Después de la instalación	142
4.3	DURA	ANTE LA	EJECUCIÓN DE KCAGENTES POR PRIMERA VEZ	.144
4.4	FUNC	CIONALI	DADES Y COMANDOS	.144
	4.4.1	Gestión	Versiones (KCVersiones)	.146
		4.4.1.1	Crear Proyecto	146
		4.4.1.2	Salvar Proyecto	147
		4.4.1.3	Abrir Proyecto	147
		4.4.1.4	Adicionar Versión	148
		4.4.1.5	Editar Versión	148
		4.4.1.6	Eliminar Versión	149
		4.4.1.7	Editar Base de Datos de una Versión	149
		4.4.1.8	Adicionar Funcionalidad a una Versión	150
		4.4.1.9	Editar una Funcionalidad	150
		4.4.1.10	Eliminar una Funcionalidad	151
			Adicionar componente a una Funcionalidad	
		4.4.1.12	Editar un Componente	152
		4.4.1.13	Eliminar un Componente	153

		4.4.1.14	Copiar Funcionalidad, Componente o Versión	153
		4.4.1.15	Pegar Funcionalidad, Componente o Versión	153
		4.4.1.16	Actualizar Agencia	153
	4.4.2	GESTI	ÓN USUARIOS (KCUsuarios)	154
		4.4.2.1	Adicionar Administrador	154
		4.4.2.2	Editar Administrador	155
		4.4.2.3	Eliminar Administrador	155
		4.4.2.4	Cambiar Sesión	156
		4.4.2.5	Monitorear Usuario	156
		4.4.2.6	Habilitar y Deshabilitar un Usuario	158
5	ANEX	OE M	ANUAL DEL PROGRAMADOR KCAGENTES	159
5.1	PREF	ACIO		159
	5.1.1	Acerca	de este Documento	159
	5.1.2	Docum	entos Relacionados	159
5.2	ARQI	JITECTI	JRA DE KCAGENTES	161
5.3	FUNC	CIONAM	IENTO DE KCAGENTES	162
	5.3.1	Gestor	Versiones	163
	5.3.2	Gestor	Persistencia	163
		5.3.2.1	Conexión a una Base de Datos	164
		5.3.2.2	Ejecución de Consultas	164
	5.3.3	Objeto	Actualizador	165
	5.3.4	Objeto	Consulta	165
	5.3.5	Gestor	Usuarios	167
	5.3.6	Interfaz	Gestión Usuarios e Interfaz Gestión Versiones	168
	5.3.7	Gestor	Configuración Cliente	168
			Aplicación e Interfaz Gráfica	
		_	La Interfaz IKCAgente	
6	ANE	OF MA	ANUAL DEL USUARIO TUTOR KCMM	170
6 1	PREF	ACIO		170
0. 1			de este Documento	
62			N	
J. <u>Z</u>			imientos	
		•	o de Instalación	
	0.2.2		Preparando la Instalación	
			Durante la Instalación	
			Después de la Instalación	

6.3	DUR	ANTE LA	A EJECUCIÓN DE KCMM POR PRIMERA VEZ	. 174
6.4	СОМ	PONEN	TES DE LA INTERFAZ	. 174
	6.4.1	Мара с	onceptual del tema o funcionalidad	. 175
	6.4.2	Control	es para la Navegación	. 178
		6.4.2.1	Control para la Navegación en la Interfaz Gráfica Analista	178
		6.4.2.2	Control para la Navegación en la Interfaz Gráfica Director de Proyectos	179
		6.4.2.3	Menú de Navegación Interfaz Gráfica Diseñador	179
		6.4.2.4	Menú de Navegación Interfaz Programador	180
	6.4.3	Consola	a para Visualizar la Información de los Diferentes Temas	. 181
	6.4.4	Ayuda .		. 181
	6.4.5	Elemen	to Lúdico	. 182
		6.4.5.1	Elemento Lúdico para la Interfaz Gráfica del Perfil Analista	182
		6.4.5.2	Elemento Lúdico para la Interfaz Gráfica del Perfil Director de Proyectos	184
		6.4.5.3	Elemento lúdico para la Interfaz Gráfica del Perfil Diseñador	184
		6.4.5.4	Elemento Lúdico para la Interfaz Gráfica del Perfil Programador	186

DIAGRAMAS DE CASOS DE USO

Diagrama Casos de Uso 1 Gestión Versiones	22
Diagrama Casos de Uso 2 Gestión Usuarios	47
Diagrama Casos de Uso 3 Tutor KCMM Versión 2.0	56

DIAGRAMAS DE SECUENCIA

Diagrama de Secuencia 1 Adicionar Componente	25
Diagrama de Secuencia 2 Modificar Componente	27
Diagrama de Secuencia 3 Eliminar Componente	29
Diagrama de Secuencia 4 Adicionar Funcionalidad	31
Diagrama de Secuencia 5 Modificar Funcionalidad	33
Diagrama de Secuencia 6 Eliminar Funcionalidad	35
Diagrama de Secuencia 7 Adicionar Versión	37
Diagrama de Secuencia 8 Modificar Versión	39
Diagrama de Secuencia 9 Eliminar Versión	41
Diagrama de Secuencia 10 Planear Actualización, Ejecutar Actualización, Actualizar Versión	45
Diagrama de Secuencia 11 Registrar Usuario	49
Diagrama de Secuencia 12 Validar Usuario	51
Diagrama de Secuencia 13 Habilitar Usuario	53
Diagrama de Secuencia 14 Modificar Usuario	55
Diagrama de Secuencia 15 Consultar Información	58
Diagrama de Secuencia 16 Evaluar Estudiante	60

DIAGRAMAS DE CLASES

Diagrama de Clases 1 Gestión Versiones	62
Diagrama de Clases 2 Gestión Usuarios	85
Diagrama de Clases 3 SW – CMM	99
Diagrama de Clases 4 Evaluación	115

LISTA DE FIGURAS

Figura 1 Representación de un objeto o agente móvil	21
Figura 2 Representación de un sistema que utiliza la tecnología de objetos y agentes móvi	les .21
Figura 3 Estados de los Agentes Grasshopper	130
Figura 4 Arquitectura Grasshopper	131
Figura 5 Conexiones de Red	140
Figura 6 Estado de Conexión de área local	140
Figura 7 Propiedades de Conexión de área local	141
Figura 8 Propiedades de Protocolo Internet (TCP/IP)	141
Figura 9 Contenido Directorio de Instalación	142
Figura 10 Contenido Carpeta Datos	143
Figura 11 Contenido Carpeta Configuración	143
Figura 12 Ejecutado KCAgentes por primera vez	144
Figura 13 Elementos de la Interfaz Gráfica	145
Figura 14 Crear Proyecto	146
Figura 15 Salvar Proyecto	147
Figura 16 Abrir Proyecto	147
Figura 17 Editar Versión	148
Figura 18 Editar Base de Datos	149
Figura 19 Adicionar Funcionalidad	150
Figura 20 Editar Funcionalidad	151
Figura 21 Editar Componente	152
Figura 22 Actualizar Agencia	153
Figura 23 Adicionar Administrador	154
Figura 24 Editar Administrador	155
Figura 25 Cambiar Sesión	156
Figura 26 Ver Datos Personales Usuario	157
Figura 27 Datos Académicos Usuario	157
Figura 28 Arquitectura del ambiente KCAgentes	161
Figura 29 Diagrama de Estados Objeto Gestor Persistencia	163
Figura 30 Diagrama de Estados Objeto Actualizador	165
Figura 31 Diagrama de Estados Objeto Consulta	166
Figura 32 Edición Gestor Persistencia en el Módulo de Gestión Versiones	167

Figura 33 Inicio - Panel de Control	171
Figura 34 Panel de Control	172
Figura 35 Red	172
Figura 36 Propiedades TCP / IP	173
Figura 37 Contenido Directorio de Instalación	173
Figura 38 Contenido Carpeta Datos	174
Figura 39 Ejecutando KCMM por primera vez	175
Figura 40 Mapa Conceptual: Conceptos Generales Sw - Cmm	175
Figura 41 Mapa Conceptual: Niveles de Sw - Cmm	176
Figura 42 Mapa Conceptual: Evaluación	176
Figura 43 Mapa Conceptual: Resultados Evaluación	177
Figura 44 Mapa Conceptual: Créditos	177
Figura 45 Control para la Navegación en la Interfaz Gráfica del Perfil Analista	178
Figura 46 Control para la Navegación en la Interfaz Gráfica del Perfil Director De Proyecto	s 179
Figura 47 Control para la Navegación en la Interfaz Gráfica del Perfil Diseñador	180
Figura 48 Control para la Navegación en la Interfaz Gráfica del Perfil Programador	180
Figura 49 Consola de Información y Mensajes	181
Figura 50 Ícono de Ayuda al Usuario	181
Figura 51 Visualización de Ayuda al Usuario	182
Figura 52 Elemento Lúdico (Rompecabezas) de la Interfaz Grafica del Perfil Analista	182
Figura 53 Botón Desordenar Mapa de la Interfaz Gráfica del Perfil Analista	183
Figura 54 Botón Armar Mapa de la Interfaz Gráfica del Perfil Analista	183
Figura 55 Elemento Lúdico (Cuadernillos de apuntes) de la Interfaz Grafica del Perfil Director	de
Proyectos	184
Figura 56 Elemento Lúdico (Herramientas) de la Interfaz Grafica Perfil Diseñador	185
Figura 57 Elemento Lúdico (Propiedades) de la Interfaz Grafica del Perfil Programador	186

1 ANEXO A DIAGRAMAS DE CASOS DE USO Y DIAGRAMAS DE SECUENCIA

OBSERVACIONES

En vista de la carencia de elementos que representen y faciliten la comprensión de sistemas que utilicen la tecnología de objetos y agentes móviles, el grupo de investigación "CORBA y OSM" dirigido por el profesor Msc. Henry Diosa propone:

La representación gráfica de un objeto o agente móvil se hace mediante la Figura 1.

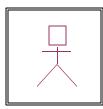


Figura 1 Representación de un objeto o agente móvil

Para la comprensión de la movilidad de objetos y agentes inherente a esta tecnología, se agrupan los casos de uso de acuerdo al espacio del sistema donde se ejecutan, visualizando la interacción y comunicación entre éstos por medio de objetos o agentes móviles (Ver Figura 2). Dado que los casos de uso utilizan los objetos o agentes móviles se estereotipa la relación bidireccional entre ellos con <<use><-use>>>.

Ejemplo: En la Figura 2, el caso de uso *Consultar Información* que se encuentra en el espacio A del sistema, interactúa por medio del Objeto Consulta (Objeto móvil), con el caso de uso *Gestionar Base de Datos* ubicado en el espacio B del sistema.

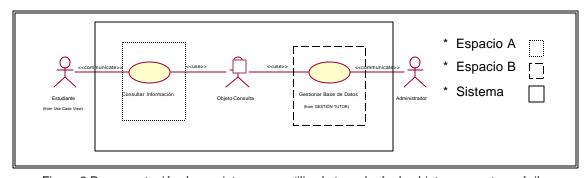


Figura 2 Representación de un sistema que utiliza la tecnología de objetos y agentes móviles

1.1 DIAGRAMAS DE CASOS DE USOS Y DIAGRAMAS DE SECUENCIAS KCAGENTES (Herramienta para la actualización de software)

1.1.1 Diagrama Gestión Versiones

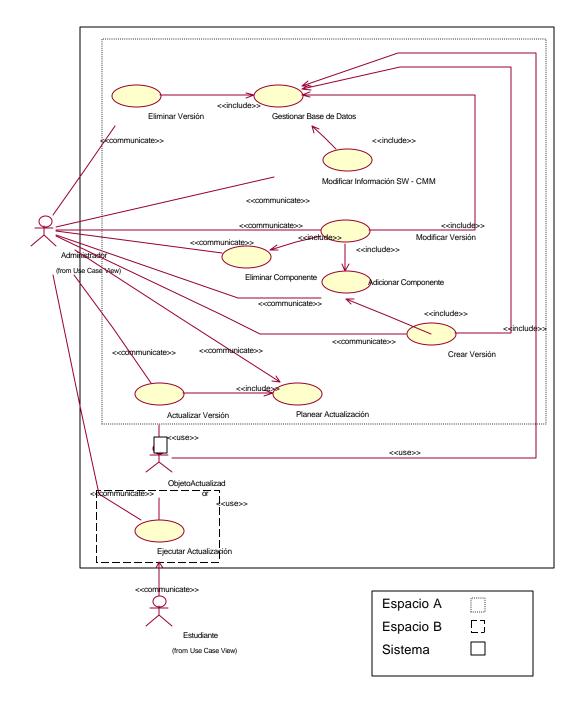
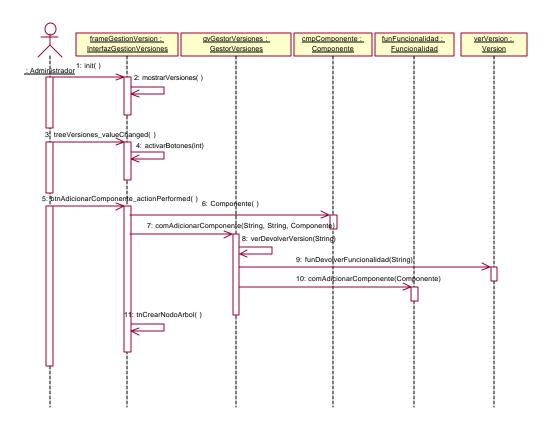


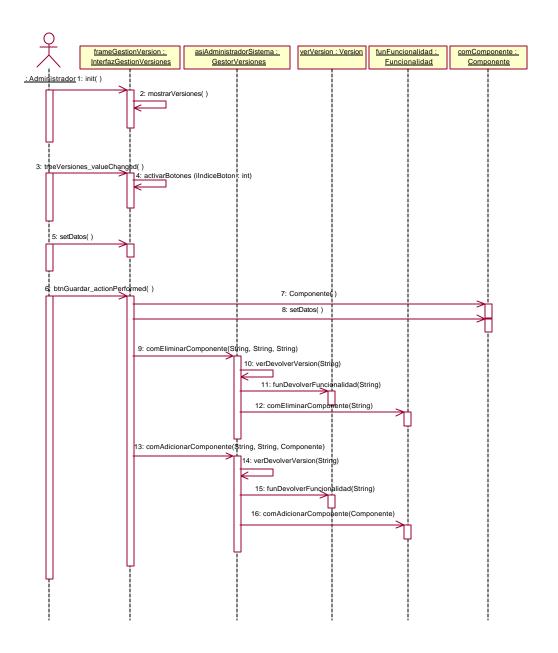
Diagrama Casos de Uso 1 Gestión Versiones

	1.1.1.1	Adicionar Componente	
PROYECTO	KCAGENTES		
CLIDCICTEMA	OFOTIÓN VEDOLONE	0	

PROYECTO	KCAGENTES	KCAGENTES						
SUBSISTEMA	GESTIÓN VERSIONES	GESTIÓN VERSIONES						
PROPÓSITO	Permite la creación de un comp	Permite la creación de un componente en el nodo administrador.						
	OBSERVACIONES							
	1. Un componente no puede	Un componente no puede crearse sin que exista al menos una versión						
	pues todos los componentes de							
	2. Los tipos de componentes so	on: interfaz gráfica	y funcionalidad					
PRIORIDAD	Primario							
ACTORES								
	ESCENARI	O NORMAL						
	ACCIÓN ACTOR	RESP	UESTA SISTEM	IA				
1. Selecciona la	Selecciona la opción de adicionar							
componente.								
		2. Solicita los dato	s del nuevo com	ponente				
		tales como: tipo de componente, nombre,						
		descripción, archivos de documentación,						
		fichero.java, fichero.class, desarrollador.						
3. Digita los dat	os solicitados por el sistema y	-						
	ión del componente.							
	·	4. Guarda la fecha de adición del componente						
		y el responsable.						
		5. Adiciona el componente, es decir copia los						
	archivos en el directorio correspondiente.							
	RELAC							
ID	CASOS DE USO	USA	INCLUYE	EXTIENDE				



1.1.1.2 Modificar Componente								
PROYE	PROYECTO KCAGENTES							
SUBSIS	SUBSISTEMA GESTIÓN VERSIONES							
PROPÓ:	PROPÓSITO Permite la modificación de un componente en el nodo administrador.							
PRIORIE	DAD	Primario						
ACTORI	ES							
		ESCENARI	0 N	IORMAL				
	Α	CCIÓN ACTOR		RESP	UESTA SISTEM	A		
1. Selec	ciona el d	componente que desea						
modificar.								
			2. Solicita los datos del componente tales			te tales		
				como: tipo de componente, nombre,				
			descripción, archivos de documentación,					
			fic	hero.java, ficher	o.class, desarroll	ador.		
3. Edita	3. Edita los datos del componente.							
4. Guarda los datos del componente.								
		RELAC	101	NES				
ID		CASOS DE USO		USA	INCLUYE	EXTIENDE		



	1.1.1.3 Eliminar Componente					
PROYECTO	KCAGENTES					
SUBSISTEMA	SUBSISTEMA GESTIÓN VERSIONES					
PROPÓSITO	Permite la eliminación de un	componente en el nodo administrador.				
	OBSERVACIONES					
	Para que este escenario esté	habilitado, debe existir en la versión actual, al				
	menos un (1) componente.					
PRIORIDAD	PRIORIDAD Primario					
ACTORES	ACTORES					
	ESCENA	RIO NORMAL				
А	CCIÓN ACTOR	RESPUESTA SISTEMA				
1. Selecciona la	funcionalidad donde se					
encuentra el con eliminar.	nponente que se desea					
		2. Lista todos los componentes existentes.				
3. Selecciona el eliminar.	componente que desea	,				
		4. Elimina el componente.				

RELACIONES

USA

INCLUYE

EXTIENDE

CASOS DE USO

ID

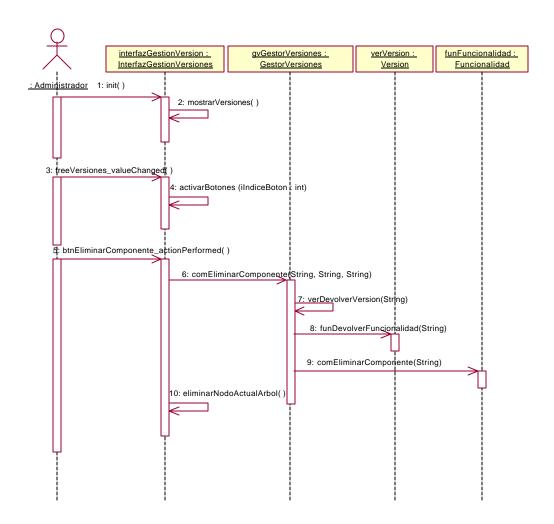
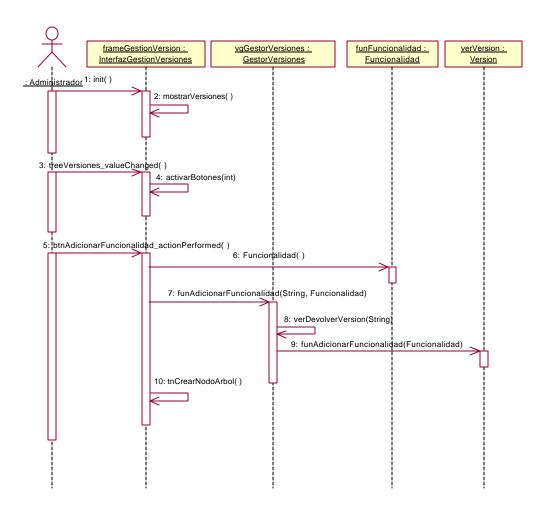


Diagrama de Secuencia 3 Eliminar Componente

1.1.1.4 Adicionar Funcionalidad

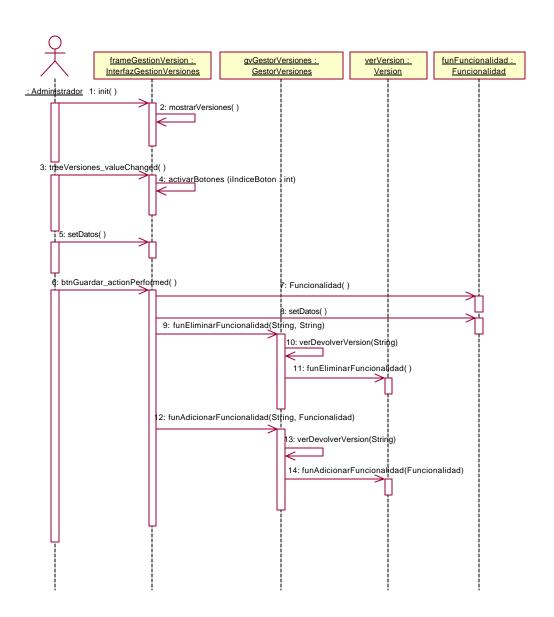
PROYEC	TO	KCAGENTES							
SUBSIST	ГЕМА	GESTIÓN VERSIONES							
PROPÓS	SITO	Permite la adición de una funcionalidad.							
		OBSERVACIONES	OBSERVACIONES						
		1. Una funcionalidad no puede	crea	arse sin que exi	sta al menos una	versión			
				-					
PRIORID	AD	Primario							
ACTORE	S								
	ESCENARIO NORMAL								
	Д	CCIÓN ACTOR		RESP	UESTA SISTEM	A			
Selecciona la opción de adicionar									
funcionali									
			2. Solicita los datos de la nueva funcionalidad						
			tales como: nombre y descripción.						
3. Digita	los dato	s solicitados por el sistema y							
confirma	la adició	n de la funcionalidad.							
				4. Guarda los datos de la funcionalidad.					
5. Adiciona la funcionalidad, es decir copia lo									
	archivos en el directorio correspondiente.								
		RELAC							
ID		CASOS DE USO		USA	INCLUYE	EXTIENDE			



1.1.1.5 Modificar Funcionalidad						
PROYECTO	KCAGENTES					
SUBSISTEMA	GESTIÓN VERSIONES					
PROPÓSITO	Permite la modificación de una funcionalidad.					
PRIORIDAD	Primario					
ACTORES						
ESCENARIO NORMAL						
ACCIÓN ACTOR		RESPUESTA SISTEMA				
Selecciona la funcionalidad que desea						
modificar.						
		2. Solicita los datos de la funcionalidad tales				
		como: nombre y descripción.				
3. Edita los datos de la funcionalidad.						
o. Eata los datos de la falloiofidilada.						
		4. Guarda los datos de la funcionalidad.				
RELACIONES						

INCLUYE EXTIENDE

CASOS DE USO USA



1.1.1.6 Eliminar Funcionalidad						
PROYECTO	KCAGENTES					
SUBSISTEMA	GESTIÓN VERSIONES					
PROPÓSITO	Permite la eliminación de un componente en el nodo administrador.					
	OBSERVACIONES					
	Para que este escenario esté habilitado, debe existir en la versión actual, al menos una (1) funcionalidad.					
PRIORIDAD	Primario					
ACTORES						
ESCENARIO NORMAL						
ACCIÓN ACTOR		RESPUESTA SISTEMA				
Selecciona la versión donde se encuentra						
la funcionalidad que se desea eliminar.						

RELACIONES

3. Selecciona la funcionalidad que desea

CASOS DE USO

eliminar.

2. Lista todas las funcionalidades existentes.

INCLUYE

EXTIENDE

4. Elimina la funcionalidad.

USA

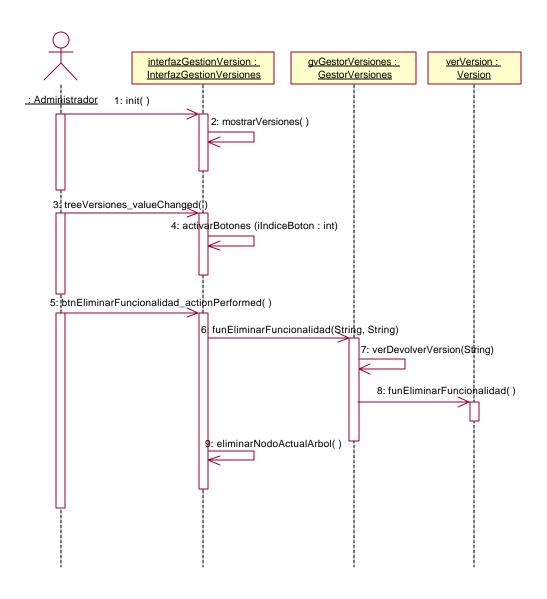


Diagrama de Secuencia 6 Eliminar Funcionalidad

1.1.1.7 Adicionar Versión

		T							
PROYEC	CTO	KCAGENTES							
SUBSIS [®]	TEMA	GESTIÓN VERSIONES							
PROPÓS	SITO	Permite la adición de una versión de una aplicación							
		'							
		OBSERVACIONES							
		Para que la versión pueda crearse, debe contener al menos una (1)							
		funcionalidad y una (1) interfaz gráfica.							
PRIORIE	DAD	Primario							
ACTORE	ES								
ESCENARIO NORMAL									
ACCIÓN ACTOR		RES	RESPUESTA SISTEMA						
Selecciona la opción de crear una versión.									
·		2. Solicita los datos de la nueva versión tales							
		como: Nombre, Descripción y Base de Datos.							
3. Digita los datos solicitados por el sistema y									
confirma la creación de la versión.									
Somma la creación de la versión.			4. Guarda la fecha de creación de la versión y						
			su responsable.						
7. Crea versión. (El sistema interpreta				reta la					
			versión, crea los directorios correspondientes a						
		la versión, y a cada una de las funcionalidades,							
			interfaces y documentación)						
RELACIONES									
ID		CASOS DE USO	USA	INCLUYE	EXTIENDE				

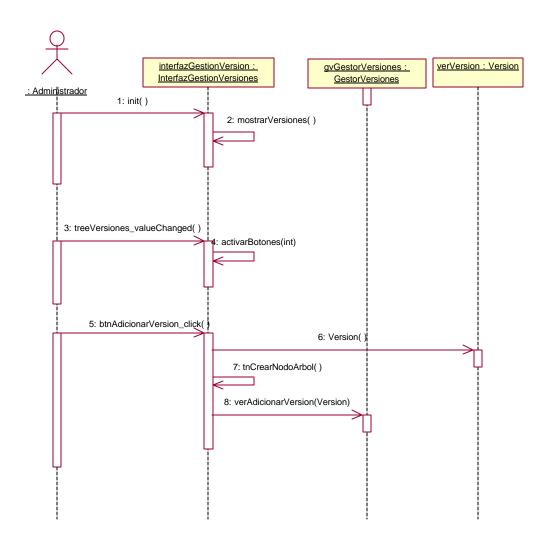


Diagrama de Secuencia 7 Adicionar Versión

1.1.1.8 Modificar Versión

PROYE	CTO	KCAGENTES				
SUBSIS	TEMA	GESTIÓN VERSIONES				
PROPÓ:	SITO	Permite la adición de una versi	ón de una aplicac	ión		
PRIORIE	DAD	Primario	•			
ACTOR	ES					
ESCENARIO NORMAL						
	Д	CCIÓN ACTOR	RESPUESTA SISTEMA			
1. Selec	ciona la	versión que desea modificar.				
			2. Solicita los da	atos de la versión	tales como:	
			Nombre, Descrip	oción y Base de D	Datos.	
3. Edita los datos solicitados por el sistema.						
			4. Guarda los da	atos de la versión.		
	RELACIONES					
ID	•	CASOS DE USO	USA	INCLUYE	EXTIENDE	

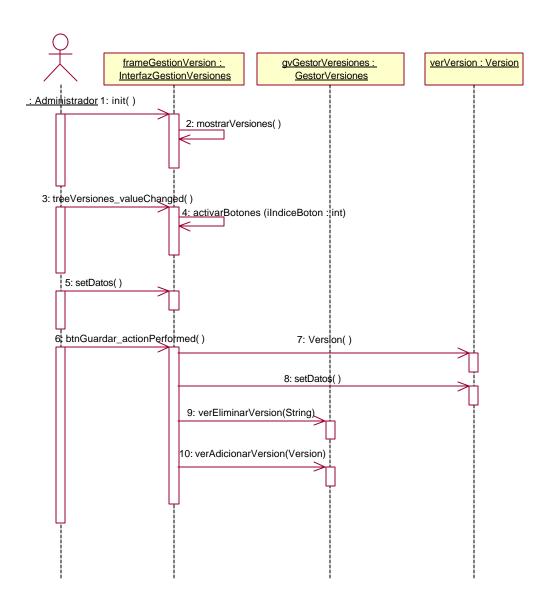


Diagrama de Secuencia 8 Modificar Versión

1.1.1.9 Eliminar Versión					
PROYECT	O	KCAGENTES			
SUBSISTE		GESTIÓN VERSIONES			
PROPÓSIT	ГО	Permite la eliminación de una v	ersión de una aplic	ación.	
		OBSERVACIONES			
		Para que este escenario esté h	abilitado, debe exi	stir al menos una	versión.
PRIORIDAI	D	Primario			
ACTORES	ACTORES				
		ESCENARIO			
	AC	CIÓN ACTOR	RESP	UESTA SISTEMA	١
1. Seleccio	na la o _l	oción de eliminar versiones.			
			2. Lista todas las versiones.		
3. Seleccio	na la ve	ersión que desea eliminar.			
4. Elimina los objetos de las agencias cliente y la versión del nodo administrador 5. Solicita Gestionar Base de Datos (Ver caso de uso 1.1.1.13)				,	
	RELACIONES				
ID		CASOS DE USO	USA	INCLUYE	EXTIENDE
1.1.1.13	Gestion	ar Base de Datos		Х	

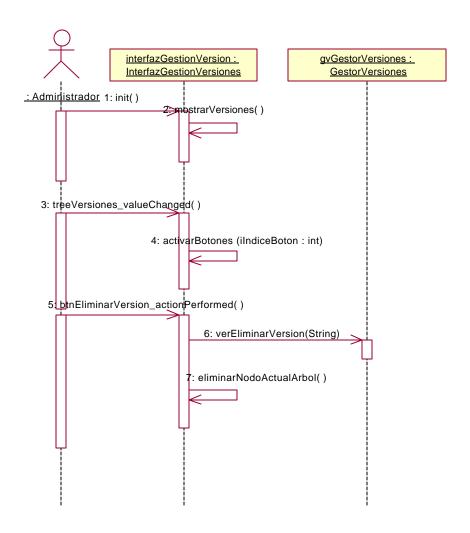


Diagrama de Secuencia 9 Eliminar Versión

	1.1.1.10 Planear Actualización				
PROYECTO	O KCAGENTES				
SUBSISTEMA	A GESTIÓN VERSIONES	GESTIÓN VERSIONES			
PROPÓSITO	Permite en la agencia adminis			que se	
	quieren actualizar y las agenc	as cliente donde se	quiere hacer la		
	actualización.				
PRIORIDAD	Primario				
ACTORES					
		IO NORMAL			
	ACCIÓN ACTOR	RESP	UESTA SISTEM	IA	
1. Selecciona	la opción planear.				
		2. Actualiza la galería de versiones.			
3. Selecciona	las versiones que desea				
actualizar en	las agencias cliente.				
		4. Muestra las age			
		el estado de las m	nismas (activas c	inactivas)	
	las agencias cliente donde				
•	realice la actualización.				
6. Confirma e	l plan de actualización.				
	7. Crea el agente actualizador con los datos				
del plan.					
	RELA	CIONES			
ID	CASOS DE USO	USA	INCLUYE	EXTIENDE	

	1.1.1.11 Ejecutar Actualización			
PROYECTO	KCAGENTES			
SUBSISTEMA	GESTIÓN VERSIONES			
PROPÓSITO	Ejecuta la actualización en las	agencias cliente.		
PRIORIDAD	Primario			
ACTORES				
	ESCENARIO	NORMAL		
ACCIÓN ACTOR RESPUESTA SISTEMA				1A
1. Termina de planear la actualización.				
3. El objeto actualizador es enviado a las agencias cliente activas. 4. El objeto actualizador es guardado en la agencias cliente activas. 5. Si existen agencias inactivas se asignar tareas pendientes al objeto actualizador. 6. Solicita Gestionar Base de Datos. (Ver caso de uso ID: 1.1.1.13)			ado en las e asignan izador.	
	RELACIO	ONES		
ID	CASOS DE USO	USA	INCLUYE	EXTIENDE
1.1.1.13 Gestiona	ar Base de Datos		X	

	1.1.1.12 Actualizar Versión				
PROYECTO	KCAGENTES				
SUBSISTEMA	GESTIÓN VERSIONES				
PROPÓSITO	ÓSITO Ejecuta una actualización				
	OBSERVACIONES				
	Para que este escenario se habilite deben haber ocurrido alguno de los siguientes casos de uso: - Crear Versión - Eliminar Versión - Modificar Versión - Adicionar Componente - Eliminar Componente				
PRIORIDAD	Primario				
ACTORES					
	ESCENARI	O NORMAL			
AC	CIÓN ACTOR	RES	PUESTA SISTEN	ЛΑ	
Selecciona la opción actualizar. Solicita Planear Actualización. (Ver Caso de uso ID: 1.1.1.10) Solicita Ejecutar actualización . (Ver Caso de uso ID: 1.1.1.11)					
	RELACI	ONES	·		
ID	CASOS DE USO	USA	INCLUYE	EXTIENDE	
1.1.1.10 Planear a	ctualización		Х		
1.1.1.11 Ejecutar a	actualización		X		

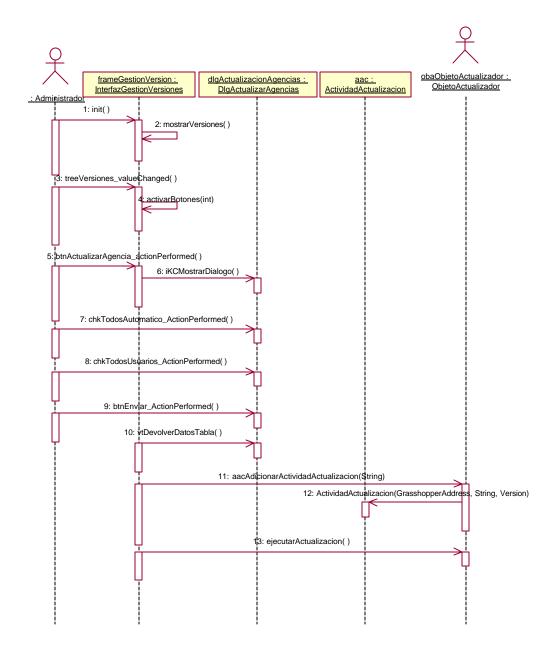


Diagrama de Secuencia 10 Planear Actualización, Ejecutar Actualización, Actualizar Versión

1.1.1.13	Gestionar Base de Datos
----------	-------------------------

PROY	ECTO	KCAGENTES				
SUBSI	ISTEMA	GESTIÓN VERSIONES				
PROP	ÓSITO	Se encarga de leer y almacena	r las estructuras d	le datos en los dis	spositivos de	
		almacenamiento. Es la funciona	alidad encargada d	de definir la maner	a como se	
		almacenan y leen los datos.				
PRIOR	RIDAD	Primario				
ACTO	RES					
		ESCENARI	O NORMAL			
	ACCIÓN ACTOR RESPUESTA SISTEMA				IA	
1. EI	Objeto Co	nsulta solicita información de				
la	base de da	atos al Sistema.				
			2. Abre la base	de datos.		
			3. Ejecuta la ser	ntencia de consult	a o	
			modificación.			
			4. Retorna la info	ormación o guarda	a las	
			modificaciones.	-		
	5. Cierra la base de datos.					
		RELAC	IONES			
ID						

1.1.2 Diagrama Gestión Usuarios

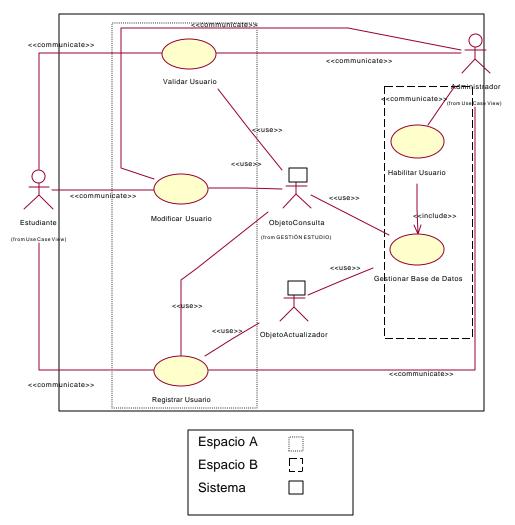


Diagrama Casos de Uso 2 Gestión Usuarios

	1.1.2.1 Reg	istrar Usuario		
PROYECTO	KCAGENTES			
SUBSISTEMA	GESTIÓN USUARIOS			
PROPÓSITO	Permite el registro de un usua	ario al sistema		
T KOI OONO	T crimic or registro de dir dode	ano ai sistema.		
	OBSERVACIONES			
	Los tipos de usuarios pued	den ser Administra	ador o estudiante	٠.
	2. El perfil del estudiante es e desarrollo de software, éstos Director.			
	Solo el administrador princi administradores.	ipal del sistema p	uede agregar nu	evos
PRIORIDAD	Primario			
ACTORES				
ESCENARIO NORMAL				
AC	ACCIÓN ACTOR RESPUESTA SISTEMA			
1. Selecciona la opción de registrar (E).				
 Solicita los siguientes datos: nombres, apellidos, correo electrónico, identificación, clave y <i>rol</i>. Digita los datos solicitados por el sistema y 				
confirma su registro	(E).			
		4. Revisa la sint	axis de los datos	
		6. Solicita Gesti	onar base de Da	tos
			reso del estudian	
	ESCENARIO E	CEPCIONAL		
AC	CIÓN ACTOR	RES	PUESTA SISTEI	MA
Selecciona la opo administrador (A).	ción de registrar un nuevo	2. Solicita los si	guientes datos: r	nombres,
apellidos, correo electrónico, identificación, clave.				ntificación,
3. Digita los datos d	lel nuevo administrador.			
		4. Revisa la sint	taxis de los datos	S.
5. Habilita el nuevo administrador para que				
		pueda ingresar	al sistema.	
,	RELACI		-	T
ID	CASOS DE USO	USA	INCLUYE	EXTIENDE
1.1.1.13 G	Sestionar base de Datos		X	

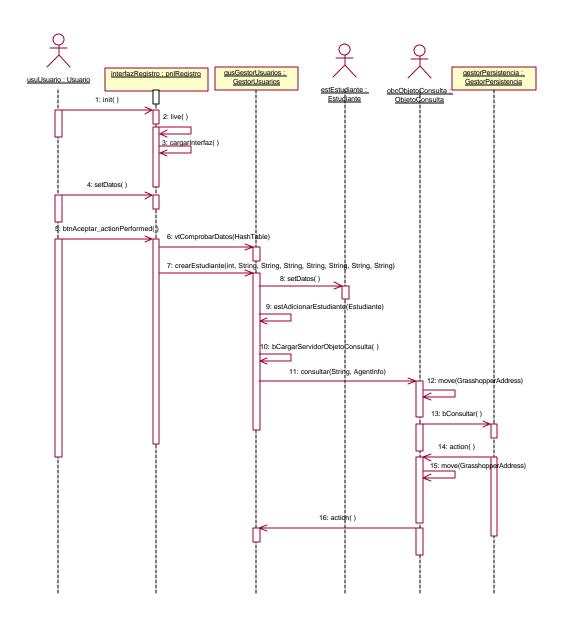


Diagrama de Secuencia 11 Registrar Usuario

1.1.2.2 Validar Usuario				
PROYECTO	KCAGENTES			
SUBSISTEMA	GESTIÓN USUARIOS			
PROPÓSITO	Permite a un usuario existente ingresar al sistema y se realiza en la agencia			
	en ejecución.			
PRIORIDAD	Primario			
ACTORES				
	ESCENARIO			
AC	CCIÓN ACTOR	RES	PUESTA SISTEN	ΛA
1. Selecciona la op	ción de ingresar (A),(E).			
		2. Solicita los sig	guientes datos: id	dentificación y
		clave.		-
3. Digita los datos	solicitados por el sistema y			
confirma su ingreso	o (A) ,(E).			
		3. Crea un obje	to consulta	
		4. Envía el obje	to consulta de la a	agencia
		cliente a la agen	cia administrador	
		5. Solicita Gestio	onar Base de Dat	os
		(Ver caso de uso	D: 1.1.1.13)	
			iltado de la consu	ılta en el
		objeto consulta.		
		•	to consulta de la a	•
			a agencia cliente	
			lave digitada con	la clave
	almacenada en la base de datos.			
9. Si las claves son iguales, permite el ingreso del estudiante al nivel de acceso que posea.				•
	DEL ACI		nivel de acceso o	que posea.
ID	CASOS DE USO	1	INCLUVE	EVTIENDE
ID Continue	r Page de Dates	USA	INCLUYE	EXTIENDE
1.1.1.13 Gestiona	r Base de Datos		X	

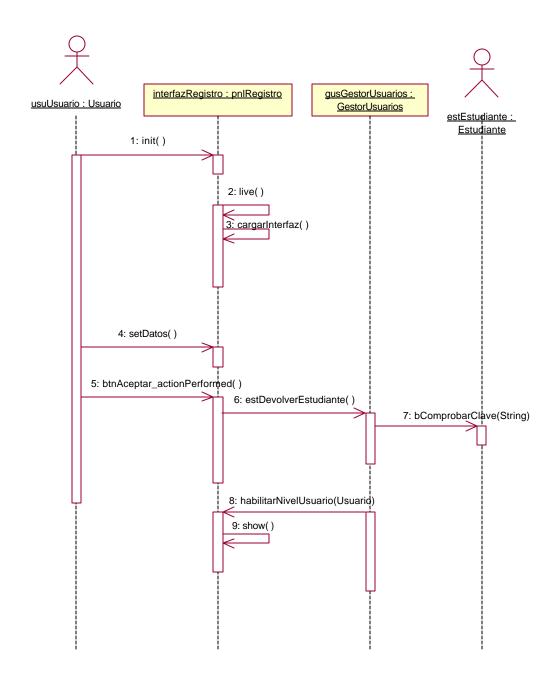


Diagrama de Secuencia 12 Validar Usuario

1.1.2.3	Habilitar Usuario

PROYECTO	KCAGENTES				
SUBSISTEN		GESTIÓN USUARIOS			
PROPÓSITO		Permite deshabilitar un usuario en el sistema desde el nodo administrador.			
	Esto implica qu consideran los	•	actualización que h pilitados.	naga en broadCas	t no se
			deshabilitar al usu ga en ese moment	•	ilita de usar
PRIORIDAD	Primario				
ACTORES					
		ESCENARI	O NORMAL		
	ACCIÓN ACTOR		RES	PUESTA SISTEM	IA
1. Seleccion	a la opción de desha	oilitar usuario.			
	•		2. Lista todos los	s usuarios.	
Seleccion deshabilitar.	3. Selecciona el usuario que desea				
			4. Deshabilita el	usuario.	
RELACIONES					
ID	CASOS DE U	SO	USA	INCLUYE	EXTIENDE

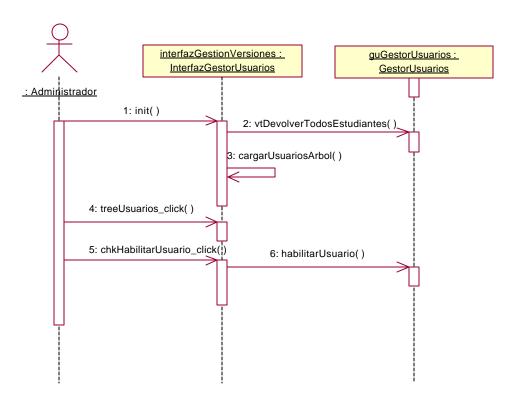


Diagrama de Secuencia 13 Habilitar Usuario

1.1.2.4 Modificar Usuario					
PROYEC	TO	KCAGENTES			
SUBSIST	ГЕМА	GESTIÓN USUARIOS			
PROPÓS	SITO	Permite la edición de datos de	el usuario.		
PRIORID	AD	Primario			
ACTORE	S				
ESCENARIO NORMAL					
ACCIÓN ACTOR RESPUESTA SISTEMA			ΛA		
Selecciona la opción de modificar datos					
(A),(E).					
			Solicita modificar los siguientes datos:		
			nombres, apellid	los, correo electró	nico, clave.
_		solicitados por el sistema y ación (A), (E).			
			4. Revisa la sin	taxis de los datos	j.
	5. Solicita Gestionar Base de Datos.				tos.
	RELACIONES				
ID		CASOS DE USO	USA	INCLUYE	EXTIENDE
1.1.1.13	Solicita G	estionar Base de Datos		Х	

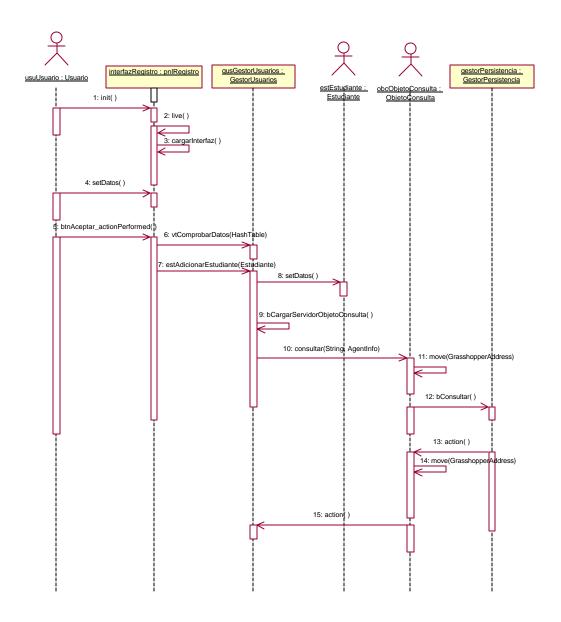


Diagrama de Secuencia 14 Modificar Usuario

1.2 DIAGRAMAS DE CASOS DE USOS Y DIAGRAMAS DE SECUENCIAS TUTOR KCMM VERSIÓN 2.0

1.2.1 Diagrama Gestión Información

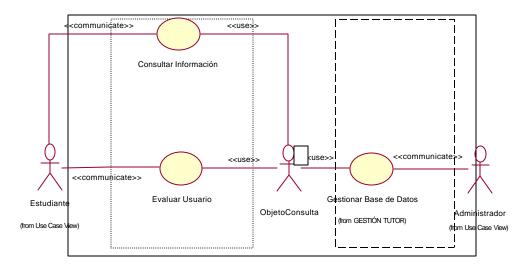


Diagrama Casos de Uso 3 Tutor KCMM Versión 2.0

	1.2.1.1 Consultar Información				
		,			
PROYEC		TUTOR KCMM VERSIÓN 2.0			
SUBSIST	ГЕМА	GESTIÓN ESTUDIO			
PROPÓS	SITO	Permite a los usuarios consul			elo de
		Capacidades de Madurez par	a Software (SW - 0	CMM).	
PRIORID	AD	Primario			
ACTORE	S				
		ESCENARIO	NORMAL		
	AC	CIÓN ACTOR	RESF	PUESTA SISTEM	IA
		ción de consultar			
informaci	ón.				
			2. Lista los niveles de Información.		
3. Selecc	ciona el nive	el de información que desea			
consultar	•				
			4. Crea el objeto consulta en la agencia		
		cliente donde se solicitó consultar			
			información.		
			5. Envía el objeto consulta de la agencia		
			cliente a la agencia administrador. 6. Solicita Gestionar Base de Datos.		
					tos.
			(Ver caso de uso		do dotoo
			7. Carga la inform		e de datos
			en el objeto consulta. 8. Envía el objeto consulta de la agencia		
			administrador a la agencia cliente.		
			9. Muestra la información.		
	ı	RELACI		10111117	EVELDE
ID		CASOS DE USO	USA	INCLUYE	EXTIENDE
1.1.1.13	Gestionar	Base de Datos		X	

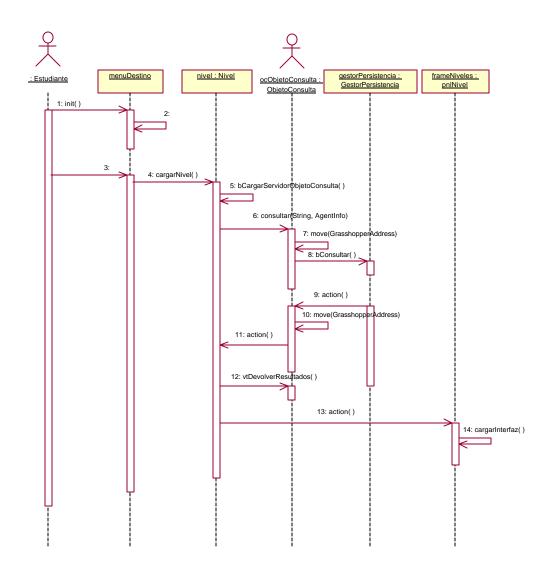


Diagrama de Secuencia 15 Consultar Información

1.2.1.2 Evaluar Estudiante

		Ia a . £			
PROYEC		TUTOR KCMM VERSIÓN 2.0			
SUBSIST	ГЕМА	GESTIÓN ESTUDIO			
PROPÓSITO Permite al estudiante evaluar		sus conocimiento	OS.		
		Como resultado le permite al	estudiante conoc	er sus debilidade	s y
		fortalezas además de propone	erle temas de estu	udio para reforzar	sus
		conocimientos respecto a SW	/ -CMM.	·	
PRIORID	AD	Primario			
ACTORE	:S	Estudiante			
		ESCENARIO	NORMAL		
	AC	CIÓN ACTOR	RES	PUESTA SISTEN	ИΑ
1. Selecc	iona la opo	ción presentar evaluación.			
		cliente donde se 3. Envía el objeccliente a la agen 4. Solicita Gesti (Ver caso de uso 5. El objeto cons con sus respues estudiar, las forta . Envía el objeto	sulta carga el cue stas correctas, lo alezas y debilidado o consulta de la a la agencia cliente	suario. agencia tos stionario, s temas a des. gencia	
7. Resuelve el cuestionario.					
		8. Captura las respuestas.			
				9. Retroalimenta al Usuario.	
_	RELACIONES				
ID		CASOS DE USO	USA	INCLUYE	EXTIENDE
1.1.1.13	Gestionar	Base de Datos		Х	

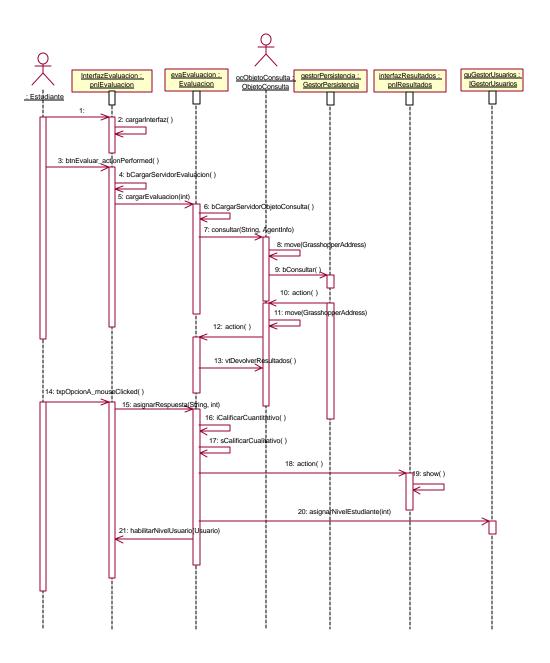


Diagrama de Secuencia 16 Evaluar Estudiante

2 ANEXO B DICCIONARIO DE CLASES

OBSERVACIONES

El formato utilizado para especificar el diccionario de clases está basado en el Lenguaje Unificado de Modelado (UML).

Los valores posibles para el campo Propiedad en los atributos son:

AddOnly: Para atributos con multiplicidad mayor a 1, especifica

que se pueden adicionar objetos pero no eliminar.

Frozen: No se puede modificar el atributo.

Changeable: No hay restricciones para modificar el atributo.

El uso de interfaces se hace necesario pues la plataforma de objetos y agentes móviles así lo exige.

DICCIONARIO DE CLASES KCAGENTE

(Herramienta para la actualización de software)

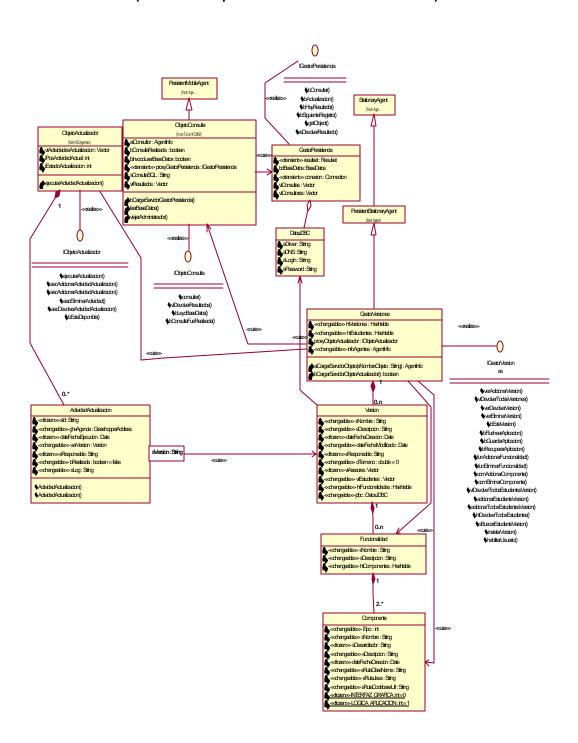


Diagrama de Clases 1 Gestión Versiones

PROYECTO	KCAgentes
SUBSISTEMA	GESTIÓN VERSIONES
VISIBILIDAD	PublicAccess
NOMBRE	1.1 IObjetoActualizador
DESCRIPCIÓN	Declara todos los servicios que debe implementar el ObjetoActualizador
PREFIJO INSTANCIA	

1.1.1 Atributos

1.1.2 Operaciones

1.1.2.1 Operación: ejecutarActualizacion

DESCRIPCIÓN	TIPO RETORNO	VISIBILIDAD
Inicia la ejecución de las actividades de		PublicAccess
actualización.		1 ublicAccess

1.1.2.2 Operación: aacAdicionarActividadActualizacion

DESCRIPCIÓN	TIPO RETORNO	VISIBILIDAD
Adiciona una actividad al objeto actualizador;		
devuelve la actividad se pudo ser adicionada, de	ActividadActualizacion	PublicAccess
lo contrario devuelve null.		

1.1.2.3 Operación: aacAdicionarActividadActualizacion

DESCRIPCIÓN	TIPO RETORNO	VISIBILIDAD
Adiciona una actividad al objeto actualizador; devuelve la actividad se pudo ser adicionada, de lo contrario devuelve null.	ActividadActualizacion	PublicAccess

1.1.2.4 Operación: aacEliminarActividad

DESCRIPCIÓN	TIPO RETORNO	VISIBILIDAD
Elimina una actividad del objeto actualizador; devuelve la actividad si pudo ser eliminada, de lo contrario devuelve null.	ActividadActualizacion	PublicAccess

1.1.2.5 Operación: aacDevolverActividadActualizacion

DESCRIPCIÓN	TIPO RETORNO	VISIBILIDAD
Devuelve una actividad del objeto actualizador; devuelve la actividad si la encuentra, de lo contrario devuelve null.	ActividadActualizacion	PublicAccess

1.1.2.6 Operación: bEstaDisponible

DESCRIPCIÓN	TIPO RETORNO	VISIBILIDAD
Devuelve true si el objetoActualizador está	boolean	PublicAccess
disponible, de lo contrario devuelve false.	Doolean	1 ublicAccess

[REPEAT16]PROYECTO	KCAGENTES
SUBSISTEMA	GESTIÓN VERSIONES
VISIBILIDAD	PublicAccess
NOMBRE	1.2 StationaryAgent
DESCRIPCIÓN	Es la clase base para aquellos agentes que no tienen permitido moverse.
	=

1.2.1 ATRIBUTOS

Ver documentación API de la plataforma para agentes móviles Grasshopper.

1.2.2 OPERACIONES

Ver documentación API de la plataforma para agentes móviles Grasshopper.

PROYECTO	KCAGENTES	
SUBSISTEMA	GESTIÓN VERSIONES	
VISIBILIDAD	PublicAccess	
NOMBRE	1.3 PersistentStationaryAgent	
DESCRIPCIÓN	Es la clase base que permite crear agentes que no desean moverse por si se	
DESCRIPCION	necesita implementar el servicio de persistencia	
PREFIJO INSTANCIA		

1.3.1 ATRIBUTOS

Ver documentación API de la plataforma para agentes móviles Grasshopper.

1.3.2 OPERACIONES

Ver documentación API de la plataforma para agentes móviles Grasshopper.

PROYECTO	KCAGENTES
SUBSISTEMA	GESTIÓN VERSIONES
VISIBILIDAD	PublicAccess
NOMBRE	1.4 ActividadActualizacion
DESCRIPCIÓN	Son cada una de las actividades que el objeto actualizador debe realizar para actualizar el software.
PREFIJO INSTANCIA	aac

1.4.1 ATRIBUTOS

1.4.1.1 Atributo: sld

DESCRIPCIÓN	Es el identificador de la actividad.

TIPO	VISIBILIDAD	PROPIEDAD	VALOR POR DEFECTO
String	PrivateAccess	frozen	

1.4.1.2 Atributo: ghaAgencia

DESCRIPCION	Guarda la PROPIEDAD de la agencia donde se desea actualizar.
-------------	--

TIPO	VISIBILIDAD	PROPIEDAD	VALOR POR DEFECTO
GrasshopperAddress	PrivateAccess	changeable	

1.4.1.3 Atributo: dateFechaEjecucion

DESCRIPCION	Guarda la fecha de la ejecución de la actividad.
-------------	--

TIPO	VISIBILIDAD	PROPIEDAD	VALOR POR DEFECTO
Date	PrivateAccess	frozen	

1.4.1.4 Atributo: verVersion

TIPO	VISIBILIDAD	PROPIEDAD	VALOR POR DEFECTO
Version	PrivateAccess	changeable	

1.4.1.5 Atributo: sResponsable

DESCRIPCIÓN	Es la identificación del usuario humano responsable de la actividad.
DESCRIPCION	i Es la identificación del usuano numano responsable de la actividad.

TIPO	VISIBILIDAD	PROPIEDAD	VALOR POR DEFECTO
String	PrivateAccess	frozen	

1.4.1.6 Atributo: bRealizada

TIPO	VISIBILIDAD	PROPIEDAD	VALOR POR DEFECTO
Boolean	PrivateAccess	changeable	false

1.4.2 OPERACIONES

1.4.2.1 Operación: ActividadActualizacion

DESCRIPCIÓN	TIPO	VISIBILIDAD
Constructor de la clase ActividadActualización.		PublicAccess

PROYECTO	KCAGENTES
SUBSISTEMA	GESTIÓN VERSIONES
VISIBILIDAD	PublicAccess
NOMBRE	1.5 IObjetoConsulta
DESCRIPCIÓN	Declara todos los servicios que debe implementar el ObjetoConsulta.
PREFIJO INSTANCIA	

1.5.1 ATRIBUTOS

1.5.2 OPERACIONES

1.5.2.1 Operación: consultar

DESCRIPCIÓN	TIPO RETORNO	VISIBILIDAD
Ejecuta una consulta en una base de datos.		PublicAccess

1.5.2.2 Operación: vtDevolverResultados

DESCRIPCIÓN	TIPO RETORNO	VISIBILIDAD
Devuelve el resultado de la consulta.	Vector	PublicAccess

1.5.2.3 Operación: bLeyoBaseDatos

DESCRIPCIÓN	TIPO RETORNO	VISIBILIDAD
Devuelve true si leyó la base de datos despues de		
haber invocado la consulta, de lo contrario	boolean	PublicAccess
devuelve false.		

1.5.2.4 Operación: bConsultaFueRealizada

DESCRIPCIÓN	TIPO RETORNO	VISIBILIDAD
Devuelve true si la consulta fue realizada, de lo		PublicAccess
contrario devuelve false.		1 abile/teeess

PROYECTO	KCAGENTES
SUBSISTEMA	GESTIÓN VERSIONES
VISIBILIDAD	PublicAccess
NOMBRE	1.6 PersistentMobileAgent
DESCRIPCIÓN	Es la clase base que permite crear agentes que no desean moverse por si se
BEGGIAII GIGIA	necesita implementar el servicio de persistencia
PREFIJO INSTANCIA	

1.6.1 ATRIBUTOS

Ver documentación API de la plataforma para agentes móviles Grasshopper.

1.6.2 OPERACIONES

Ver documentación API de la plataforma para agentes móviles Grasshopper.

PROYECTO	KCAGENTES
SUBSISTEMA	GESTIÓN VERSIONES
VISIBILIDAD	PublicAccess
NOMBRE	1.7 Funcionalidad
DESCRIPCIÓN	 -> Es un conjunto de componentes que realizan tareas para lograr un objetivo común. -> Una funcionalidad debe tener al menos un (1) componente de lógica de aplicación y un (1) componente de interfaz gráfica.
PREFIJO INSTANCIA	fun

1.7.1 ATRIBUTOS

1.7.1.1 Atributo: sNombre

DESCRIPCIÓN	Es el nombre que identifica la funcionalidad.

TIPO	VISIBILIDAD	PROPIEDAD	VALOR POR DEFECTO
String	PrivateAccess	Changeable	

1.7.1.2 Atributo: sDescripcion

DESCRIPCION	Es una breve descripción de la funcionalidad.

TIPO	VISIBILIDAD	PROPIEDAD	VALOR POR DEFECTO
String	PrivateAccess	changeable	

1.7.1.3 Atributo: htComponentes

TIPO	VISIBILIDAD	PROPIEDAD	VALOR POR DEFECTO
Hashtable	PrivateAccess	changeable	

1.7.2 OPERACIONES

1.7.2.1 Operación: Funcionalidad

DESCRIPCIÓN		TIPO RETORNO	VISIBILIDAD
Constructor de la clase funcionalidad.			PublicAccess

1.7.2.2 Operación: comAdicionarComponente

DESCRIPCIÓN	TIPO RETORNO	VISIBILIDAD
-> Asigna un componente a la funcionalidad. No admite componentes con igual nombre; en caso de		
que así sea la edita.	Componente	PublicAccess
-> Devuelve el componente si pudo ser asignado		
de lo contrario devuelve null.		

1.7.2.3 Operación: setDatos

DESCRIPCIÓN	TIPO RETORNO	VISIBILIDAD
Asigna los valores a los atributos de una funcionalidad.		PublicAccess

1.7.2.4 Operación: getDatos

DESCRIPCIÓN	TIPO RETORNO	VISIBILIDAD
Obtiene los valores de los atributos de una		PublicAccess
funcionalidad.		1 ublicAccess

1.7.2.5 Operación: comEliminarComponente

1	DESCRIPCIÓN	TIPO RETORNO	VISIBILIDAD
	Elimina un componente de la funcionalidad;		
	devuelve el componente si pudo ser eliminado de	Componente	PublicAccess
	lo contrario devuelve null.		

1.7.2.6 Operación: comDevolverComponente

DESCRIPCIÓN	TIPO RETORNO	VISIBILIDAD
Devuelve el componente especificado por su nombre.	Componente	PublicAccess

1.7.2.7 Operación: vtDevolverTodosComponentes

DESCRIPCIÓN	TIPO RETORNO	VISIBILIDAD
Devuelve todos los componentes de la versión	Vector	PublicAccess

1.7.2.8 Operación: bEstaComponente

DESCRIPCIÓN	TIPO RETORNO	VISIBILIDAD
Verifica que un componente exista.	boolean	PublicAccess

PROYECTO	KCAGENTES
SUBSISTEMA	GESTIÓN VERSIONES
VISIBILIDAD	PublicAccess
NOMBRE	1.8 Componente
DESCRIPCIÓN	 -> Es la unidad básica de una versión, que contine los datos necesarios para la creación de un agente en una plataforma de agentes. -> Ellos proveen servicios que pueden agruparse en una funcionalidad. -> Existen dos tipos de componentes: - 0 = Interfaz gráfica - 1 = Lógica de aplicación
PREFIJO INSTANCIA	com

1.8.1 ATRIBUTOS

1.8.1.1 Atributo: iTipo

DESCRIPCIÓN	Define el tipo de componente: - Interfaz Gráfica = 0
	- Lógica de aplicación = 1

TIPO	VISIBILIDAD	PROPIEDAD	VALOR POR DEFECTO
Int	PrivateAccess	changeable	

1.8.1.2 Atributo: sNombre

DESCRIPCIÓN	Guarda el nombre del componente.
-------------	----------------------------------

TIPO	VISIBILIDAD	PROPIEDAD	VALOR POR DEFECTO
String	PrivateAccess	changeable	

1.8.1.3 Atributo: sDesarrollador

DESCRIPCION	Guarda el nombre del desarrollador del componente

TIPO	VISIBILIDAD	PROPIEDAD	VALOR POR DEFECTO
String	PrivateAccess	frozen	

1.8.1.4 Atributo: sDescripcion

DESCRIPCION	Es una breve descripción del componente

TIPO	VISIBILIDAD	PROPIEDAD	VALOR POR DEFECTO
String	PrivateAccess	changeable	

1.8.1.5 Atributo: dateFechaCreacion

DESCRIPCIÓN	Guarda la fecha de creación del componente dentro del sistema.

TIPO	VISIBILIDAD	PROPIEDAD	VALOR POR DEFECTO
Date	PrivateAccess	frozen	

1.8.1.6 Atributo: sRutaClassName

TIPO	VISIBILIDAD	PROPIEDAD	VALOR POR DEFECTO
String	PrivateAccess	changeable	

1.8.1.7 Atributo: sRutaJava

TIPO	VISIBILIDAD	PROPIEDAD	VALOR POR DEFECTO
String	PrivateAccess	changeable	

1.8.1.8 Atributo: sRutaCodebaseUrl

DESCRIPCIÓN Guarda el URL de las clases que utiliza el componente.	
---	--

TIPO	VISIBILIDAD	PROPIEDAD	VALOR POR DEFECTO
String	PrivateAccess	changeable	

1.8.1.9 Atributo: INTERFAZ_GRAFICA

DESCRIPCIÓN	Constante que especifica el tipo de componente: Interfaz Gráfica

TIPO	VISIBILIDAD	PROPIEDAD	VALOR POR DEFECTO
Int	PrivateAccess	frozen	0

1.8.1.10 Atributo: LOGICA_APLICACION

DESCRIPCION	Constante que especifica el tipo de componente: Lógica de Aplicación
-------------	--

TIPO	VISIBILIDAD	PROPIEDAD	VALOR POR DEFECTO
Int	PrivateAccess	frozen	1

1.8.2 OPERACIONES

1.8.2.1 Operación: Componente

DESCRIPCIÓN	TIPO RETORNO	VISIBILIDAD
Constructor de la clase		PublicAccess

1.8.2.2 Operación: setDatos

DESCRIPCIÓN	TIPO RETORNO	VISIBILIDAD
Asigna los valores a los atributos de un		PublicAccess
componente.		T abile/ (cccss

1.8.2.3 Operación: bEsInterfazGrafica

DESCRIPCIÓN	TIPO RETORNO	VISIBILIDAD
Retorna true si el componente es una interfaz	boolean	PublicAccess
gráfica, de lo contrario devuelve false.	booloan	1 451107 (00000

1.8.2.4 Operación: getDatos

DESCRIPCIÓN	TIPO RETORNO	VISIBILIDAD
Obtiene los valores de los atributos de un componente.		PublicAccess

PROYECTO	KCAGENTES
SUBSISTEMA	GESTION VERSIONES
VISIBILIDAD	PublicAccess
NOMBRE	1.9 ObjetoActualizador
DESCRIPCIÓN	Es un objeto de software móvil, que solicita los servicios necesarios para la
DEGORII GIGIN	actualización del sistema, tanto al lado cliente como en el lado del servidor.
PREFIJO INSTANCIA	oba

1.9.1 ATRIBUTOS

1.9.1.1 Atributo: vtActividadesActualizacion

DESCRIPCIÓN	Es una colección de las actividades que el objeto actualizador debe realizar.
-------------	---

TIPO	VISIBILIDAD	PROPIEDAD	VALOR POR DEFECTO
Vector	PrivateAccess		

1.9.1.2 Atributo: iPosActividadActual

TIPO	VISIBILIDAD	PROPIEDAD	VALOR POR DEFECTO
Int	PrivateAccess		

1.9.1.3 Atributo: iEstadoActualizacion

DESCRIPCIÓN	Representa el estado del objeto actualizador:
DESCRIPCION	- Inactivo = 0 - Actualizando = 1

TIPO	VISIBILIDAD	PROPIEDAD	VALOR POR DEFECTO
Int	PrivateAccess		

1.9.2 OPERACIONES

1.9.2.1 Operación: ejecutarActividadActualizacion

DESCRIPCIÓ	N	TIPO RETORNO	VISIBILIDAD
Ejecuta la actividad actual. PrivateAccess		PrivateAccess	

PROYECTO	KCAGENTES
SUBSISTEMA	GESTIÓN VERSIONES
VISIBILIDAD	PublicAccess
NOMBRE	1.10 Version
DESCRIPCIÓN	Es un conjunto de funcionalidades que constituyen el producto que se le entrega a los usuarios para satisfacer sus requerimientos.
PREFIJO INSTANCIA	ver

1.10.1 ATRIBUTOS

1.10.1.1 Atributo: sNombre

DESCRIPCIÓN	Guarda el nombre que identifica la versión.

TIPO	VISIBILIDAD	PROPIEDAD	VALOR POR DEFECTO
String	PrivateAccess	changeable	

1.10.1.2 Atributo: sDescripcion

TIPO	VISIBILIDAD	PROPIEDAD	VALOR POR DEFECTO
String	PrivateAccess	changeable	

1.10.1.3 Atributo: dateFechaCreacion

DESCRIPCION	Guarda la	ı fecha de creación de l	a versión en el	sistema.
-------------	-----------	--------------------------	-----------------	----------

TIPO	VISIBILIDAD	PROPIEDAD	VALOR POR DEFECTO
Date	PrivateAccess	frozen	

1.10.1.4 Atributo: dateFechaModificado

DESCRIPCION	Define la fecha de modificación de la versión en el sistema.

TIPO	VISIBILIDAD	PROPIEDAD	VALOR POR DEFECTO
Date	PrivateAccess	changeable	

1.10.1.5 Atributo: sResponsable

DESCRIPCION	
	Guarda el responsable de la versión.

TIPO	VISIBILIDAD	PROPIEDAD	VALOR POR DEFECTO
String	PrivateAccess	frozen	

1.10.1.6 Atributo: dTamano

TIPO	VISIBILIDAD	PROPIEDAD	VALOR POR DEFECTO
Double	PrivateAccess	changeable	0

1.10.1.7 Atributo: vtAsesores

TIPO	VISIBILIDAD	PROPIEDAD	VALOR POR DEFECTO
Vector	PrivateAccess	frozen	

1.10.1.8 Atributo: vtEstudiantes

DESCRIPCIÓN	Guarda los nombres de los estudiantes de la versión.

TIPO	VISIBILIDAD	PROPIEDAD	VALOR POR DEFECTO
Vector	PrivateAccess	changeable	

1.10.1.9 Atributo: htFuncionalidades

DESCRIPCION	Guarda los nombres de los estudiantes de la versión.
-------------	--

TIPO	VISIBILIDAD	PROPIEDAD	VALOR POR DEFECTO
Hashtable	PrivateAccess	changeable	

1.10.2 OPERACIONES

1.10.2.1 Operación: Version

DESCRIPCIÓN	TIPO RETORNO	VISIBILIDAD
Constructor de la clase Versión.		PublicAccess

1.10.2.2 Operación: setDatos

DESCRIPCIÓN	TIPO RETORNO	VISIBILIDAD
Asigna los valores a los atributos de una versión.		PublicAccess

1.10.2.3 Operación: getDatos

DESCRIPCIÓN	TIPO RETORNO	VISIBILIDAD
Obtiene los valores de los atributos de una versión.		PublicAccess

1.10.2.4 Operación: eliminarComponente

DESCRIPCIÓN	TIPO RETORNO	VISIBILIDAD
Elimina un componente de una versión.	boolean	PublicAccess

1.10.2.5 Operación: funAdicionarFuncionalidad

DESCRIPCION	TIPO RETORNO	VISIBILIDAD
Adiciona una funcionalidad a una versión.	Funcionalidad	PublicAccess

1.10.2.6 Operación: funDevolverFuncionalidad

DESCRIPCIÓN	TIPO RETORNO	VISIBILIDAD
Devuelve una funcionalidad especificada por su		
nombre, si esta se encuentra de lo contrario	Funcionalidad	PublicAccess
devuelve null.		

1.10.2.7 Operación: funEliminarFuncionalidad

DESCRIPCIÓN	TIPO RETORNO	VISIBILIDAD
Elimina una Funcionalidad	Funcionalidad	PublicAccess

1.10.2.8 Operación: vtDevolverTodosComponentes

DESCRIPCIÓN	TIPO RETORNO	VISIBILIDAD
Devuelve todos los componentes de una versión.	Vector	PublicAccess

1.10.2.9 Operación: vtDevolverTodasFuncionalidades

DESCRIPCIÓN	TIPO RETORNO	VISIBILIDAD
Devuelve todas las funcionalidades de una versión.	Vector	PublicAccess

1.10.2.10 Operación: vtDevolverTodosEstudiantes

DESCRIPCIÓN	TIPO RETORNO	VISIBILIDAD
Devuelve todos los nombres de los estudiantes de la versión.	Vector	PublicAccess

1.10.2.11 Operación: vtDevolverAs esores

DESCRIPCIÓN	TIPO RETORNO	VISIBILIDAD
Devuelve todos los asesores de la versión.	Vector	PublicAccess

1.10.2.12 Operación: sBuscarEstudiante

DESCRIPCIÓN	TIPO RETORNO	VISIBILIDAD
Devuelve la identificación del estudiante.	String	PublicAccess

PROYECTO	KCAGENTES
SUBSISTEMA	GESTIÓN VERSIONES
VISIBILIDAD	PublicAccess
NOMBRE	1.11 ObjetoConsulta
DESCRIPCIÓN	 -> Es un objeto de software móvil que solicita servicios de acceso a datos, tanto al lado del servidor como el lado del cliente. -> Este componente debe viajar entre el nodo cliente y el nodo servidor para realizar sus operaciones y solicitarle algunos servicios. -> En el nodo servidor es el único componente que interactúa con el objeto gestorPersistencia, quien es el guardián de la base de datos.
PREFIJO INSTANCIA	Obc

1.11.1 ATRIBUTOS

1.11.1.1 Atributo: aiConsultor

	Guarda la información del objeto que solicitó el servicio de consulta.
DESCRIPCIÓN	Cuando ningún objeto ha solicitado el servicio o el servicio fue prestado, este atributo toma el valor de null.

TIPO	VISIBILIDAD	PROPIEDAD	VALOR POR DEFECTO
AgentInfo	PrivateAccess		

1.11.1.2 Atributo: bConsultaRealizada

DESCRIPCIÓN	Devuelve el valor true si la consulta fue realizada, de lo contrario devuelve false.
-------------	--

TIPO	VISIBILIDAD	PROPIEDAD	VALOR POR DEFECTO
Bolean	PrivateAccess		

1.11.1.3 Atributo: blnvocoLeerBaseDatos

DESCRIPCIÓN	Devuelve el valor true si se invocó leer la base de datos, de lo contrario es	
DESCRIPCION	false.	

TIPO	VISIBILIDAD	PROPIEDAD	VALOR POR DEFECTO
Bolean	PrivateAccess		

1.11.1.4 Atributo: proxyGestorPersistencia

DESCRIPCION Es una referencia al objeto gestorPersistencia.	
--	--

TIPO	VISIBILIDAD	PROPIEDAD	VALOR POR DEFECTO
IGestorPersistencia	PrivateAccess	transient	

1.11.1.5 Atributo: sConsultaSQL

	Guarda la consulta que se va a solicitar al objeto consulta.
DESCRIPCIÓN	Cuando no existen consultas o la consulta ya fue realizada este atributo toma el valor de null, de lo contrario es un String.

TIPO	VISIBILIDAD	PROPIEDAD	VALOR POR DEFECTO
String	PrivateAccess		

1.11.1.6 Atributo: vtResultados

DESCRIPCIÓN	Guarda el resultado de la consulta proveniente del resultset del objeto gestor
DEGGIAN GIGIT	persistencia.

TIPO	VISIBILIDAD	PROPIEDAD	VALOR POR DEFECTO
Vector	PrivateAccess		

1.11.2 OPERACIONES

1.11.2.1 Operación: bCargarServidorGestorPersistencia

DESCRIPCIÓN	TIPO RETORNO	VISIBILIDAD
Busca el gestorPersistencia y carga una referencia.		
Devuelve true si el gestorPersistencia fue cargado satisfactoriamente, de lo contrario devuelve false.	boolean	PrivateAccess

1.11.2.2 Operación: leerBaseDatos

DESCRIPCIÓN	TIPO RETORNO	VISIBILIDAD
Invoca el gestorPersistencia para que lea la base de datos.		PrivateAccess

1.11.2.3 Operación: viajarAdministrador

DESCRIPCIÓN	TIPO RETORNO	VISIBILIDAD
Permite viajar al objeto consulta a la agencia administradora.		PrivateAccess

PROYECTO	KCAGENTES
SUBSISTEMA	GESTIÓN VERSIONES
VISIBILIDAD	PublicAccess
NOMBRE	1.12 IRegion
DESCRIPCIÓN	Es la clase que permite gestionar a todos los agentes registrados en una region
PREFIJO INSTANCIA	

1.12.1 ATRIBUTOS

Ver documentación API de la plataforma para agentes móviles Grasshopper.

1.12.2 OPERACIONES

Ver documentación API de la plataforma para agentes móviles Grasshopper.

PROYECTO	KCAGENTES
SUBSISTEMA	GESTIÓN VERSIONES
VISIBILIDAD	PublicAccess
NOMBRE	1.13 GestorVersiones
DESCRIPCIÓN	Administra las versiones y usuarios existentes en el sistema.
PREFIJO INSTANCIA	asi

1.13.1 ATRIBUTOS

1.13.1.1 Atributo: htVersiones

DESCRIPCIÓN	Contiene la colección de versiones del sistema.
-------------	---

TIPO	VISIBILIDAD	PROPIEDAD	VALOR POR DEFECTO
Hashtable	PrivateAccess	changeable	

1.13.1.2 Atributo: htEstudiantes

DESCRIPCION	Contiene la colección de estudiantes del sistema.
-------------	---

TIPO	VISIBILIDAD	PROPIEDAD	VALOR POR DEFECTO
Hashtable	PrivateAccess	changeable	

1.13.1.3 Atributo: proxyObjetoActualizador

DESCRIPCION	l Carraga vafavanaja al abiata astropina dav
I DESCRIPCION	l Es una referencia al objeto actualizador.

TIPO	VISIBILIDAD	PROPIEDAD	VALOR POR DEFECTO
IObjetoActualizador	PrivateAccess		

1.13.1.4 Atributo: infoAgentes

DECODIDATA	
DESCRIPCION	l Contiene información de los agentes que se buscarán en la agencia.

TIPO	VISIBILIDAD	PROPIEDAD	VALOR POR DEFECTO
AgentInfo	PrivateAccess	changeable	

1.13.2 OPERACIONES

1.13.2.1 Operación: aiCargarServidorObjeto

DESCRIPCIÓN	TIPO RETORNO	VISIBILIDAD
Carga la información de un agente específico.	AgentInfo	PrivateAccess

1.13.2.2 Operación: bCargarServidorObjetoActualizador

DESCRIPCIÓN	TIPO RETORNO	VISIBILIDAD
Carga el proxy del objeto actualizador; devuelve true si lo encontró, de lo contrario devuelve false.	boolean	PrivateAccess

PROYECTO	KCAGENTES
SUBSISTEMA	GESTIÓN VERSIONES
VISIBILIDAD	PublicAccess
NOMBRE	1.14 IGestorVersiones
DESCRIPCIÓN	Declara todos los servicios que debe implementar el objeto GestorVersiones
PREFIJO INSTANCIA	

1.14.1 ATRIBUTOS

1.14.2 OPERACIONES

1.14.2.1 Operación: verAdicionarVersion

DESCRIPCIÓN	TIPO RETORNO	VISIBILIDAD
Adiciona una versión al sistema; devuelve la versión si pudo ser adicionada, de lo contrario devuelve null.	Version	PublicAccess

1.14.2.2 Operación: vtDevolverTodasVersiones

DESCRIPCIÓN	TIPO RETORNO	VISIBILIDAD
Devuelve todas las versiones existentes en el	Vector	PublicAccess
sistema.		

1.14.2.3 Operación: verDevolverVersion

DESCRIPCIÓN	TIPO RETORNO	VISIBILIDAD
Devuelve una versión del sistema; devuelve la versión si pudo encontrarla, de lo contrario devuelve null.	Version	PublicAccess

1.14.2.4 Operación: verEliminarVersion

DESCRIPCIÓN	TIPO RETORNO	VISIBILIDAD
Elimina una versión del sistema; devuelve la versión si puede ser eliminada, de lo contrario devuelve null.	Version	PublicAccess

1.14.2.5 Operación: bEstaVersion

DESCRIPCIÓN	TIPO RETORNO	VISIBILIDAD
Devuelve true si la versión se encuentra, de lo contrario devuelve false.	boolean	PublicAccess

1.14.2.6 Operación: funAdicionarFuncionalidad

DESCRIPCIÓN	TIPO RETORNO	VISIBILIDAD
Adiciona una funcionalidad a una versión; devuelve la funcionalidad si la encuentra, de lo contrario retorna null.	Funcionalidad	PublicAccess

1.14.2.7 Operación: funEliminarFuncionalidad

DESCRIPCIÓN	TIPO RETORNO	VISIBILIDAD
Elimina una funcionalidad de una versión; devuelve la funcionalidad se pudo ser eliminada, de lo contrario devuelve null.	Funcionalidad	PublicAccess

1.14.2.8 Operación: comAdicionarComponente

DESCRIPCIÓN		TIPO RETORNO	VISIBILIDAD
	Adiciona un componente a una funcionalidad.	Componente	PublicAccess

1.14.2.9 Operación: comEliminarComponente

DESCRIPCIÓN	TIPO RETORNO	VISIBILIDAD
Elimina un componente de una funcionalidad; devuelve el componente si pudo ser eliminado, de	Componente	PublicAccess
lo contrario devuelve null.		

1.14.2.10 Operación: vtDevolverTodosEstudiantesVersion

DESCRIPCIÓN	TIPO RETORNO	VISIBILIDAD
Devuelve todos los estudiantes de una versión.	Vector	PublicAccess

1.14.2.11 Operación: adicionarEstudianteVersion

DESCRIPCIÓN	TIPO RETORNO	VISIBILIDAD
Adiciona un estudiante a una versión.	Estudiante	PublicAccess

1.14.2.12 Operación: adicionarTodosEstudiantesVersion

DESCRIPCIÓN TIPO RETORNO VISIBIL		VISIBILIDAD
Adiciona todos los estudiantes de una versión.	PublicAccess	

1.14.2.13 Operación: instalarVersion

DESCRIPCIÓN	TIPO RETORNO	VISIBILIDAD
Instala una versión en el nodo cliente.	PublicAccess	

GESTIÓN USUARIOS

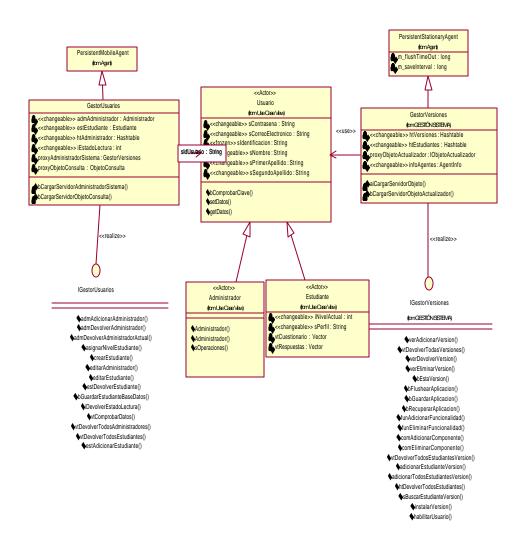


Diagrama de Clases 2 Gestión Usuarios

PROYECTO	KCAGENTES
SUBSISTEMA	Use Case View
VISIBILIDAD	PublicAccess
NOMBRE	1.15 Administrador
DESCRIPCIÓN	Es el usuario encargado de administrar el sistema. Puede realizar las siguientes operaciones: - Gestionar las versiones, funcionalidades y componentes de software Gestionar los usuarios - Monitorear el estado de cada uno de los agentes del sistema
PREFIJO INSTANCIA	Adm

1.16 ATRIBUTOS

1.17 OPERACIONES

1.17.1.1 Operación: Administrador

DESCRIPCIÓN	TIPO RETORNO	VISIBILIDAD
Constructor de la clase		PublicAccess

1.17.1.2 Operación: sOperaciones

DESCRIPCIÓN	TIPO RETORNO	VISIBILIDAD
Almacena las operaciones que realiza el	String	PublicAccess
administrador.	Stillig	1 dblicAccess

1.17.1.3 Operación: asignarOperacion

DESCRIPCIÓN	TIPO RETORNO	VISIBILIDAD
Asigna una operación a un administrador		PublicAccess

1.17.1.4 Operación: sDevolverOperacion

DESCRIPCIÓN	TIPO RETORNO	VISIBILIDAD
Devuelve una operación realizada por el	String	PublicAccess
administrador.	Stillig	1 451107 (00000

PROYECTO	KCAGENTES
SUBSISTEMA	GESTIÓN VERSIONES
VISIBILIDAD	PublicAccess
NOMBRE	1.18 PersistentStationaryAgent
DESCRIPCIÓN	Es la clase base para aquella que permite crear agentes que no desean moverse por si necesita implementar el servicio de persistencia
PREFIJO INSTANCIA	

1.19 ATRIBUTOS

Ver documentación API de la plataforma para agentes móviles Grasshopper.

1.20 OPERACIONES

Ver documentación API de la plataforma para agentes móviles Grasshopper.

PROYECTO	KCAGENTES
SUBSISTEMA	Use Case View
VISIBILIDAD	PublicAccess
NOMBRE	1.21 IGestorVersiones
DESCRIPCIÓN	Declara todos los servicios que debe implementar el objeto GestorVersiones
PREFIJO INSTANCIA	

1.22 ATRIBUTOS

1.23 OPERACIONES

1.23.1.1 Operación: verAdicionarVersion

I	DESCRIPCIÓN	TIPO RETORNO	VISIBILIDAD
	Adiciona un versión al sistema; devuelve la versión si pudo ser adicionada, de lo contrario devuelve null.	Version	PublicAccess

1.23.1.2 Operación: vtDevolverTodasVersiones

DESCRIPCIÓN	TIPO RETORNO	VISIBILIDAD
Devuelve todas las versiones existentes en el	Vector	PublicAccess
sistema.	VCCIO	1 abile/ (cccss

1.23.1.3 Operación: verDevolverVersion

DESCRIPCIÓN	TIPO RETORNO	VISIBILIDAD
Devuelve una versión del sistema; devuelve la versión si pudo encontrarla, de lo contrario devuelve null.	Version	PublicAccess

1.23.1.4 Operación: verEliminarVersion

DESCRIPCIÓN	TIPO RETORNO	VISIBILIDAD
Elimina una versión del sistema; devuelve la versión si puede ser eliminada, de lo contrario devuelve null.	Version	PublicAccess

1.23.1.5 Operación: bEstaVersion

DESCRIPCIÓN	TIPO RETORNO	VISIBILIDAD
Devuelve true si la versión se encuentra, de lo contrario devuelve false.	boolean	PublicAccess

1.23.1.6 Operación: funAdicionarFuncionalidad

DESCRIPCIÓN	TIPO RETORNO	VISIBILIDAD
Adiciona una funcionalidad a una versión; devuelve la funcionalidad si la encuentra, de lo contrario retorna null.	Funcionalidad	PublicAccess

1.23.1.7 Operación: funEliminarFuncionalidad

DESCRIPCIÓN	TIPO RETORNO	VISIBILIDAD
Elimina una funcionalidad de una versión; devuelve la funcionalidad se pudo ser eliminada, de lo contrario devuelve null.	Funcionalidad	PublicAccess

1.23.1.8 Operación: comAdicionarComponente

DESCRIPCIÓN	TIPO RETORNO	VISIBILIDAD
Adiciona un componente a una funcionalidad.	Componente	PublicAccess

1.23.1.9 Operación: comEliminarComponente

DESCRIPCIÓN	TIPO RETORNO	VISIBILIDAD
Elimina un componente de una funcionalidad; devuelve el componente si pudo ser eliminado, de	Componente	PublicAccess
lo contrario devuelve null.		

1.23.1.10 Operación: vtDevolverTodosEstudiantesVersion

DESCRIPCIÓN	TIPO RETORNO	VISIBILIDAD
Devuelve todos los estudiantes de una versión.	Vector	PublicAccess

1.23.1.11 Operación: adicionarEstudianteVersion

DESCRIPCIÓN	TIPO RETORNO	VISIBILIDAD
Adiciona un estudiante a una versión.	Estudiante	PublicAccess

1.23.1.12 Operación: adicionarTodosEstudiantesVersion

DESCRIPCIÓN	TIPO RETORNO	VISIBILIDAD
Adiciona todos los estudiantes de una versión.		PublicAccess

1.23.1.13 Operación: instalarVersion

DESCRIPCIÓN	TIPO RETORNO	VISIBILIDAD
Instala una versión en el nodo cliente.		PublicAccess

PROYECTO	KCAGENTES
SUBSISTEMA	Use Case View
VISIBILIDAD	PublicAccess
NOMBRE	1.24 Estudiante
DESCRIPCIÓN	SW - CMM es un modelo de calidad para el desarrollo de software, por lo cual, las personas que hacen parte del equipo de desarrollo de software en una empresa, son los usuarios potenciales del módulo 1 del Tutor KCAgentes. Los cuatro (4) perfiles identificados en este ámbito y a los cuales se les diseñarán sus escenarios son: - Analista - Diseñador - Programador - Director de Proyectos
PREFIJO INSTANCIA	Est

1.25 ATRIBUTOS

1.25.1.1 Atributo: iNivelActual

DESCRIPCIÓN	Es el nivel actual del estudiante. Puede ser: - 0 = Conceptos Generales - 1 = Nivel Inicial - 2 = Nivel Repetible - 3 = Nivel Definido - 4 = Nivel Gestionado
	- 5 = Nivel Optimizado

TIPO	VISIBILIDAD	PROPIEDAD	VALOR POR DEFECTO
Int	PrivateAccess	changeable	

1.25.1.2 Atributo: sPerfil

DESCRIPCIÓN	Es el perfil del estudiante. Puede ser: - Analista - Diseñador - Programador - Director de Proyector
-------------	--

TIPO	VISIBILIDAD	PROPIEDAD	VALOR POR DEFECTO
String	PrivateAccess	changeable	

1.25.1.3 Atributo: vtCuestionario

TIPO	VISIBILIDAD	PROPIEDAD	VALOR POR DEFECTO
Vector	PrivateAccess		

1.25.1.4 Atributo: vtRespuestas

DESCRIPCION	Almacena las respuestas de un cuestionario.
-------------	---

TIPO	VISIBILIDAD	PROPIEDAD	VALOR POR DEFECTO
Vector	PrivateAccess		

1.26 OPERACIONES

1.26.1.1 Operación: asignarCuestionario

DESCRIPCIÓN	TIPO RETORNO	VISIBILIDAD
Asigna un cuestionario a la colección.		PublicAccess

1.26.1.2 Operación: asignarRespuestas

DESCRIPCIÓN	TIPO RETORNO	VISIBILIDAD
Asigna las respuestas de un cuestionario.		PublicAccess

1.26.1.3 Operación: Estudiante

DESCRIPCIÓN	TIPO RETORNO	VISIBILIDAD
Constructor		PublicAccess

1.26.1.4 Operación: Estudiante

DESCRIPCIÓN	TIPO RETORNO	VISIBILIDAD
Constructor		PublicAccess

1.26.1.5 Operación: vtDevolverCuestionario

DESCRIPCIÓN	TIPO RETORNO	VISIBILIDAD
Devuelve un cuestionario.	Vector	PublicAccess

1.26.1.6 Operación: vtDevolverRespuestas

DESCRIPCIÓN	TIPO RETORNO	VISIBILIDAD
Devuelve las respuestas de un cuestionario, de lo contrario retorna null.	Vector	PublicAccess

PROYECTO	KCAGENTES
SUBSISTEMA	Use Case View
VISIBILIDAD	PublicAccess
NOMBRE	1.27 IGestorUsuarios
DESCRIPCIÓN	Declara los servicios necesarios para poder gestionar a los usuarios.
PREFIJO INSTANCIA	

1.27.1 ATRIBUTOS

1.27.2 OPERACIONES

1.27.2.1 Operación: admAdicionarAdministrador

DESCRIPCIÓN	TIPO RETORNO	VISIBILIDAD
Adiciona un administrador en la base de datos y en		
la lógica de aplicación; devuelve un administrador		PublicAccess
si pudo ser adicionado, de lo contrario retorna null.		

1.27.2.2 Operación: admDevolverAdministrador

DESCRIPCIÓN	TIPO RETORNO	VISIBILIDAD
Devuelve el administrador si es encontrado, de lo contrario devuelve null.	Administrador	PublicAccess

1.27.2.3 Operación: admDevolverAdministradorActual

DESCRIPCIÓN	TIPO RETORNO	VISIBILIDAD
Devuelve el administrador actual si es encontrado,	Administrador	PublicAccess
de lo contrario devuelve null.	Administracor	1 ublicAccess

1.27.2.4 Operación: asignarNivelEstudiante

DESCRIPCIÓN	TIPO RETORNO	VISIBILIDAD
Asigna el nivel al estudiante.		PublicAccess

1.27.2.5 Operación: crearEstudiante

DESCRIPCIÓN	TIPO RETORNO	VISIBILIDAD
Crea un estudiante en la base de datos y en la		PublicAccess
lógica de aplicación.		1 ubilicaccess

1.27.2.6 Operación: estDevolverEstudiante

DESCRIPCIÓN	TIPO RETORNO	VISIBILIDAD
Devuelve un estudiante del sistema; devuelve el estudiante si lo encuentra, de lo contrario devuelve	Estudiante	PublicAccess
null.		

1.27.2.7 Operación: estAdicionarEstudiante

DESCRIPCIÓN	TIPO RETORNO	VISIBILIDAD
Adiciona un estudiante al sistema; devuelve el estudiante si pudo ser adicionado de lo contrario	Estudiante	PublicAccess
devuelve null.	Litudiante	T ubilicAccess

1.27.2.8 Operación: htDevolverTodosEstudiantes

DESCRIPCIÓN	TIPO RETORNO	VISIBILIDAD
Devuelve todos los estudiantes del sistema.	Hashtable	PublicAccess

1.27.2.9 Operación: editarAdministrador

DESCRIPCIÓN	TIPO RETORNO	VISIBILIDAD
Permite modificar los datos personales de un administrador.		PublicAccess

1.27.2.10 Operación: editarEstudiante

DESCRIPCIÓN	TIPO RETORNO	VISIBILIDAD
Edita los datos de un estudiante.		PublicAccess

1.27.2.11 Operación: estDevolverEstudiante

DESCRIPCIÓN	TIPO RETORNO	VISIBILIDAD
Devuelve un estudiante actual.	Estudiante	PublicAccess

1.27.2.12 Operación: bGuardarEstudianteBaseDatos

DESCRIPCIÓN	TIPO RETORNO	VISIBILIDAD
Almacena el estudiante en la base de datos.	boolean	PublicAccess

1.27.2.13 Operación: iDevolverEstadoLectura

DESCRIPCIÓN	TIPO RETORNO	VISIBILIDAD
Devuelve el estado de lectura del estudiante en la	int	PublicAccess
base de datos.	l III C	1 abileAccess

1.27.2.14 Operación: vtComprobarDatos

DESCRIPCIÓN	TIPO RETORNO	VISIBILIDAD
Comprueba los datos de un usuario	Vector	PublicAccess

1.27.2.15 Operación: vtDevolverTodosAdministradores

DESCRIPCIÓN	TIPO RETORNO	VISIBILIDAD
Devuelve todos los administradores del sistema.	Vector	PublicAccess

PROYECTO	KCAGENTES
SUBSISTEMA	Use Case View
VISIBILIDAD	PublicAccess
NOMBRE	1.28 GestorUsuarios
DESCRIPCIÓN	Permite gestionar toda la información de los usuarios.

1.28.1 ATRIBUTOS

1.28.1.1 Atributo: admAdministrador

DESCRIPCIÓN	Es el administrador del sistema
-------------	---------------------------------

TIPO	VISIBILIDAD	PROPIEDAD	VALOR POR DEFECTO
Administrador	PrivateAccess	changeable	

1.28.1.2 Atributo: estEstudiante

DESCRIPCIÓN Es un estudiante del sistema.

TIPO	VISIBILIDAD	PROPIEDAD	VALOR POR DEFECTO
Estudiante	PrivateAccess	changeable	

1.28.1.3 Atributo: htAdministrador

DESCRIPCION	Guarda los datos de los administradores.
-------------	--

TIPO	VISIBILIDAD	PROPIEDAD	VALOR POR DEFECTO
Hashtable	PrivateAccess	changeable	

1.28.1.4 Atributo: iEstadoLectura

DESCRIPCIÓN Es el estado de lectura de la base de datos

TIPO	VISIBILIDAD	PROPIEDAD	VALOR POR DEFECTO
Int	PrivateAccess	changeable	

1.28.1.5 Atributo: proxyGestorVersiones

DESCRIPCIÓN Es una referencia al Administrador del sistema.

TIPO	VISIBILIDAD	PROPIEDAD	VALOR POR DEFECTO
GestorVersiones	PrivateAccess		

1.28.1.6 Atributo: proxyObjetoConsulta

DESCRIPCION	Es una referencia al objeto consulta.
-------------	---------------------------------------

TIPO	VISIBILIDAD	PROPIEDAD	VALOR POR DEFECTO
ObjetoConsulta	PrivateAccess		

1.28.2 OPERACIONES

1.28.2.1 Operación: bCargarServidorGestorVersiones

DESCRIPCIÓN	TIPO RETORNO	VISIBILIDAD
Devuelve true si se pudo cargar el servidor administrador del sistema, de lo contrario retorna null.	boolean	PrivateAccess

1.28.2.2 Operación: bCargarServidorObjetoConsulta

DESCRIPCIÓN	TIPO RETORNO	VISIBILIDAD
Devuelve true si se pudo cargar el servidor objeto consulta, de lo contrario retorna null.	boolean	PrivateAccess

PROYECTO	KCAGENTES
SUBSISTEMA	Use Case View
VISIBILIDAD	PublicAccess
NOMBRE	1.29 GestorVersiones
DESCRIPCIÓN	Administra las versiones y usuarios existentes en el sistema.
PREFIJO INSTANCIA	asi

1.29.1 ATRIBUTOS

1.29.1.1 Atributo: htVersiones

DESCRIPCIÓN	Contiene la colección de versiones del sistema.
DECOINI CICIT	Contiene la colección de versiones del sistema.

TIPO	VISIBILIDAD	PROPIEDAD	VALOR POR DEFECTO
Hashtable	PrivateAccess	changeable	

1.29.1.2 Atributo: htEstudiantes

DESCRIPCION	Contiene la colección de estudiantes del sistema.
-------------	---

TIPO	VISIBILIDAD	PROPIEDAD	VALOR POR DEFECTO
Hashtable	PrivateAccess	changeable	

1.29.1.3 Atributo: proxyObjetoActualizador

DESCRIPCIÓN	Es una referencia al objeto actualizador.

TIPO	VISIBILIDAD	PROPIEDAD	VALOR POR DEFECTO
IObjetoActualizador	PrivateAccess		_

1.29.1.4 Atributo: infoAgentes

DESCRIPCIÓN	Contiene información de los agentes que se buscarán en la agencia.
DECCIVII CICIA	Contierie información de los agentes que se buscaran en la agencia.

TIPO	VISIBILIDAD	PROPIEDAD	VALOR POR DEFECTO
AgentInfo	PrivateAccess	changeable	

1.29.2 OPERACIONES

1.29.2.1 Operación: aiCargarServidorObjeto

DESCRIPCIÓN	TIPO RETORNO	VISIBILIDAD
Carga la información de un agente específico.	AgentInfo	PrivateAccess

1.29.2.2 Operación: bCargarServidorObjetoActualizador

DESCRIPCIÓN	TIPO RETORNO	VISIBILIDAD
Carga el proxy del objeto actualizador; devuelve	boolean	PrivateAccess
true si lo encontró, de lo contrario devuelve false.	boolean	1 Hvate/teeess

PROYECTO	KCAGENTES
SUBSISTEMA	Use Case View
VISIBILIDAD	PublicAccess
NOMBRE	1.30 Usuario
DESCRIPCIÓN	Es la representación de un usuario del sistema
PREFIJO INSTANCIA	Usu

1.30.1 ATRIBUTOS

1.30.1.1 Atributo: sContrasena

DESCRIPCIÓN	Es la contraseña del usuario.

TIPO	VISIBILIDAD	PROPIEDAD	VALOR POR DEFECTO
String	PrivateAccess	changeable	

1.30.1.2 Atributo: sCorreoElectronico

DESCRIPCIÓN	Es el correo electrónico del usuario.
-------------	---------------------------------------

TIPO	VISIBILIDAD	PROPIEDAD	VALOR POR DEFECTO
String	PrivateAccess	changeable	

1.30.1.3 Atributo: sldentificacion

DESCRIPCION	Es la identificación del usuario.
-------------	-----------------------------------

TIPO	VISIBILIDAD	PROPIEDAD	VALOR POR DEFECTO
String	PrivateAccess	frozen	

1.30.1.4 Atributo: sNombre

DESCRIPCIÓN Es el nombre del usuario.

TIPO	VISIBILIDAD	PROPIEDAD	VALOR POR DEFECTO
String	PrivateAccess	changeable	

1.30.1.5 Atributo: sPrimerApellido

DESCRIPCION	Es el primer apellido del usuario.	
-------------	------------------------------------	--

TIPO	VISIBILIDAD	PROPIEDAD	VALOR POR DEFECTO
String	PrivateAccess	changeable	

1.30.1.6 Atributo: sSegundoApellido

DESCRIPCIÓN	Es el segundo apellido del usuario.

TIPO	VISIBILIDAD	PROPIEDAD	VALOR POR DEFECTO
String	PrivateAccess	changeable	

1.30.2 OPERACIONES

1.30.2.1 Operación: bComprobarClave

DESCRIPCIÓN	TIPO RETORNO	VISIBILIDAD
Devuelve true si la contraseña del usuario es	boolean	PublicAccess
correcta de lo contrario devuelve false.	Boologii	1 45110/100000

1.30.2.2 Operación: setDatos

DESCRIPCIÓN	TIPO RETORNO	VISIBILIDAD
Asigna los valores a los atributos de un usuario.		PublicAccess

1.30.2.3 Operación: getDatos

DESCRIPCIÓN	TIPO RETORNO	VISIBILIDAD
Obtiene los valores de los atributos de un usuario.		PublicAccess

DICCIONARIO DE CLASES TUTOR KCMM VERSIÓN 2.0 (Caso de Estudio)

SW - CMM

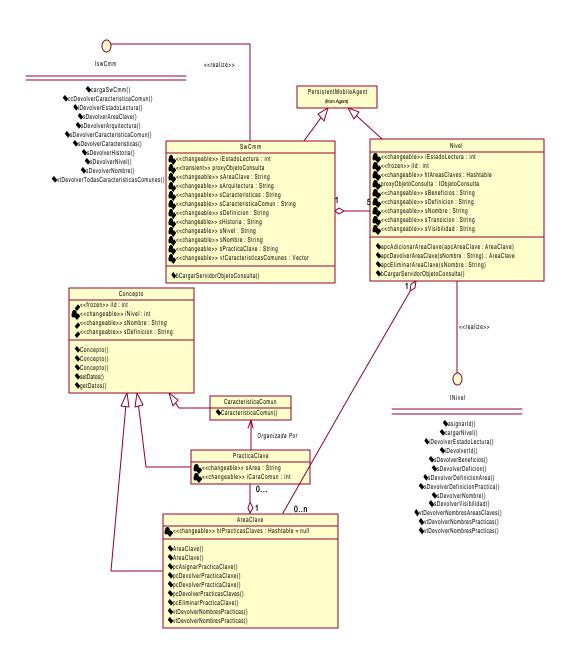


Diagrama de Clases 3 SW - CMM

PROYECTO	KCMM VERSIÓN 2.0
SUBSISTEMA	GESTIÓN TUTOR
VISIBILIDAD	PublicAccess
NOMBRE	1.31 Concepto
DESCRIPCIÓN	Es una superclase raiz de la que se derivan cada uno de los elementos que componen el modelo SW - CMM.
PREFIJO INSTANCIA	cpto

1.31.1 ATRIBUTOS

1.31.1.1 Atributo: ild

DESCRIPCIÓN	Guarda el identificador del concepto.

TIPO	VISIBILIDAD	PROPIEDAD	VALOR POR DEFECTO
Int	PublicAccess	frozen	

1.31.1.2 Atributo: iNivel

DESCRIPCION	Guarda el nivel al cual está asociado el concepto.
-------------	--

TIPO	VISIBILIDAD	PROPIEDAD	VALOR POR DEFECTO
Int	PrivateAccess	changeable	

1.31.1.3 Atributo: sNombre

TIPO	VISIBILIDAD	PROPIEDAD	VALOR POR DEFECTO
String	PublicAccess	changeable	

1.31.1.4 Atributo: sDefinicion

DESCRIPCION	Es un valor alfanumérico que contiene la definición del concepto.
-------------	---

TIPO	VISIBILIDAD	PROPIEDAD	VALOR POR DEFECTO
String	PublicAccess	changeable	

1.31.2 OPERACIONES

1.31.2.1 Operación: Concepto

DESCRIPCIÓN	TIPO RETORNO	VISIBILIDAD
Constructor de la clase Concepto.		PublicAccess

1.31.2.2 Operación: setDatos

DESCRIPCIÓN	TIPO RETORNO	VISIBILIDAD
Asigna los valores a los atributos de la clase.	Object	PublicAccess

1.31.2.3 Operación: getDatos

DESCRIPCIÓN	TIPO RETORNO	VISIBILIDAD
Obtiene los valores de los atributos de la clase.	Object	PublicAccess

PROYECTO	KCMM VERSION 2.0
SUBSISTEMA	GESTIÓN TUTOR
VISIBILIDAD	PublicAccess
NOMBRE	1.32 CaracteristicaComun
DESCRIPCIÓN	Es una abstracción de una característica común del modelo SW - CMM.
PREFIJO INSTANCIA	ccn

1.32.1 ATRIBUTOS

1.32.2 OPERACIONES

1.32.2.1 Operación: CaracteristicaComun

DESCRIPCIÓN	TIPO RETORNO	VISIBILIDAD
Constructor de la clase CarcteristicaComun		PublicAccess

PROYECTO	KCMM VERSIÓN 2.0
SUBSISTEMA	GESTIÓN TUTOR
VISIBILIDAD	PublicAccess
NOMBRE	1.33 PersistentMobileAgent
DESCRIPCIÓN	Es la clase base que permite crear agentes se pueden mover por si se necesita implementar el servicio de persistencia
PREFIJO INSTANCIA	

1.33.1 ATRIBUTOS

Ver documentación API de la plataforma para agentes móviles Grasshopper.

1.33.2 OPERACIONES

Ver documentación API de la plataforma para agentes móviles Grasshopper.

PROYECTO	KCMM VERSIÓN 2.0
SUBSISTEMA	GESTIÓN TUTOR
VISIBILIDAD	PublicAccess
NOMBRE	1.34 PracticaClave
DESCRIPCIÓN	Es una abstracción de una práctica clave de proceso en SW – CMM
PREFIJO INSTANCIA	рср

1.34.1 ATRIBUTOS

1.34.1.1 Atributo: sArea

DESCRIPCIÓN	Es la variable que almacena el identificador nombre de la área clave asociada	
DESCRIFCION	a una práctica clave.	

TIPO	VISIBILIDAD	PROPIEDAD	VALOR POR DEFECTO
String	PrivateAccess	changeable	

1.34.1.2 Atributo: iCaraComun

DESCRIPCIÓN	Variable que almacena el identificador (id) de la característica común
DESCRIPCION	asociada a la práctica clave.

TIPO	VISIBILIDAD	PROPIEDAD	VALOR POR DEFECTO
Int	PrivateAccess	changeable	

1.34.2 OPERACIONES

1.34.2.1 Operación: PracticaClave

DESCRIPCIÓN	TIPO RETORNO	VISIBILIDAD
Constructor		PublicAccess

PROYECTO	KCMM VERSION 2.0
SUBSISTEMA	GESTION TUTOR
VISIBILIDAD	PublicAccess
NOMBRE	1.35 INivel
DESCRIPCIÓN	Declara todos los servicios que implementa la clase Nivel.

1.35.1 ATRIBUTOS

1.35.2 OPERACIONES

1.35.2.1 Operación: asignarld

DESCRIPCIÓN	TIPO RETORNO	VISIBILIDAD
Asigna el identificador de un nivel.		PublicAccess

1.35.2.2 Operación: cargarNivel

DESCRIPCIÓN	TIPO RETORNO	VISIBILIDAD
Carga un nivel de madurez.		PublicAccess

1.35.2.3 Operación: iDevolverEstadoLectura

DESCRIPCIÓN	TIPO RETORNO	VISIBILIDAD
Devuelve el estado de lectura de la base de datos de SW – CMM.	int	PublicAccess

1.35.2.4 Operación: iDevolverId

DESCRIPCIÓN	TIPO RETORNO	VISIBILIDAD
Devuelve el identificador de un nivel.	int	PublicAccess

1.35.2.5 Operación: sDevolverBeneficios

DESCRIPCIÓN	TIPO RETORNO	VISIBILIDAD
Devuelve los beneficios del Nivel.	String	PublicAccess

1.35.2.6 Operación: sDevolverDeficion

DESCRIPCIÓN	TIPO RETORNO	VISIBILIDAD
Devuelve la definición del nivel.	String	PublicAccess

1.35.2.7 Operación: sDevolverDefinicionArea

DESCRIPCIÓN	TIPORETORNO	VISIBILIDAD
Devuelve la definición del área clave si es	String	PublicAccess
encontrada, de lo contrario devuelve null.	String	1 ublicAccess

1.35.2.8 Operación: sDevolverDefinicionPractica

DESCRIPCIÓN	TIPO RETORNO	VISIBILIDAD
Devuelve la definición de una práctica clave.	String	PublicAccess

1.35.2.9 Operación: sDevolverNombre

DESCRIPCIÓN	TIPO RETORNO	VISIBILIDAD
Devuelve el nombre del nivel.	String	PublicAccess

1.35.2.10 Operación: sDevolverVisibilidad

DESCRIPCIÓN	TIPO RETORNO	VISIBILIDAD
Devuelve la visibilidad del nivel.	String	PublicAccess

1.35.2.11 Operación: vtDevolverNombresAreasClaves

DESCRIPCIÓN	TIPO RETORNO	VISIBILIDAD
Devuelve una colección de todos los nombres de las áreas claves.	Vector	PublicAccess

1.35.2.12 Operación: vtDevolverNombresPracticas

DESCRIPCIÓN	TIPO RETORNO	VISIBILIDAD
Devuelve una colección con el nombre de las	Vector	PublicAccess
prácticas claves.	Vector	1 abilio/ toccss

1.35.2.13 Operación: vtDevolverNombresPracticas

DESCRIPCIÓN	TIPO RETORNO	VISIBILIDAD
Devuelve una colección con los nombres de las	Vector	PublicAccess
prácticas claves.	V 00101	1 45110/100000

PROYECTO	KCMM VERSIÓN 2.0
SUBSISTEMA	GESTIÓN TUTOR
VISIBILIDAD	PublicAccess
NOMBRE	1.36 AreaClave
DESCRIPCIÓN	Es una abstracción de área clave de proceso del modelo SW - CMM.
PREFIJO INSTANCIA	аср

1.36.1 ATRIBUTOS

1.36.1.1 Atributo: htPracticasClaves

DESCRIPCIÓN	Almacena las prácticas claves.
-------------	--------------------------------

TIPO	VISIBILIDAD	PROPIEDAD	VALOR POR DEFECTO
Hashtable	PrivateAccess	changeable	null

1.36.2 OPERACIONES

1.36.2.1 Operación: AreaClave

DESCRIPCIÓN	TIPO RETORNO	VISIBILIDAD
Constructor de la clase AreaClave.		PublicAccess

1.36.2.2 Operación: pcAsignarPracticaClave

DESCRIPCIÓN	TIPO RETORNO	VISIBILIDAD
Asigna una práctica clave.		PublicAccess

1.36.2.3 Operación: pcDevolverPracticaClave

DESCRIPCIÓN	TIPO RETORNO	VISIBILIDAD
Devuelve una práctica clave.	PracticaClave	PublicAccess

1.36.2.4 Operación: pcDevolverPracticaClave

DESCRIPCIÓN	TIPO RETORNO	VISIBILIDAD
Devuelve una práctica clave.	PracticaClave	PublicAccess

1.36.2.5 Operación: pcDevolverPracticasClaves

DESCRIPCIÓN	TIPO RETORNO	VISIBILIDAD
Devuelve las prácticas claves asociadas a una característica común.	PracticaClave	PublicAccess

1.36.2.6 Operación: pcEliminarPracticaClave

DESCRIPCIÓN	TIPO RETORNO	VISIBILIDAD
Elimina un práctica clave.	PracticaClave	PublicAccess

1.36.2.7 Operación: vtDevolverNombresPracticas

DESCRIPCIÓN	TIPO RETORNO	VISIBILIDAD
Devuelve los nombres de las prácticas claves.	Vector	PublicAccess

PROYECTO	KCMM VERSION 2.0
SUBSISTEMA	GESTION TUTOR
VISIBILIDAD	PublicAccess
NOMBRE	1.37 SwCmm
DESCRIPCIÓN	Es una abstracción de los conceptos generales del modelo de madurez de capacidades SW - CMM.
PREFIJO INSTANCIA	cmm

1.37.1 ATRIBUTOS

1.37.1.1 Atributo: iEstadoLectura

	Devuelve el estado de lectura de SW-CMM	
DESCRIPCIÓN	-1 = Init. 0 = Leyendo conceptos generales. 1 = Conceptos generales leídos, leyendo características	comunes.
	-2 = Listo.	

	TIPO	VISIBILIDAD	PROPIEDAD	VALOR POR DEFECTO
Int		PrivateAccess	changeable	

1.37.1.2 Atributo: proxyObjetoConsulta

2200 ma rotoronola ar objeto concuta.	DESCRIPCIÓN	Es una referencia al objeto consulta.
---------------------------------------	-------------	---------------------------------------

	TIPO	VISIBILIDAD	PROPIEDAD	VALOR POR DEFECTO
I		PrivateAccess	transient	

1.37.1.3 Atributo: sAreaClave

DESCRIPCIÓN	Es la descripción de área clave.
-------------	----------------------------------

TIPO	VISIBILIDAD	PROPIEDAD	VALOR POR DEFECTO
String	PrivateAccess	changeable	

1.37.1.4 Atributo: sArquitectura

TIPO	VISIBILIDAD	PROPIEDAD	VALOR POR DEFECTO
String	PrivateAccess	changeable	

1.37.1.5 Atributo: sCaracteristicas

DESCRIPCION	Es la descripción de las características de SW - CMM.
-------------	---

TIPO	VISIBILIDAD	PROPIEDAD	VALOR POR DEFECTO
String	PrivateAccess	changeable	

1.37.1.6 Atributo: sCaracteristicaComun

DESCRIPCION Es la descripción de las características comunes de SW - CMM.	
--	--

TIPO	VISIBILIDAD	PROPIEDAD	VALOR POR DEFECTO
String	PrivateAccess	changeable	

1.37.1.7 Atributo: sDefinicion

DESCRIPCIÓN	Es la descripción de la definición de SW - CMM.

TIPO	VISIBILIDAD	PROPIEDAD	VALOR POR DEFECTO
String	PrivateAccess	changeable	

1.37.1.8 Atributo: sHistoria

DESCRIPCIÓN Es la descripción de la historia de SW - CMM.	
--	--

TIPO	VISIBILIDAD	PROPIEDAD	VALOR POR DEFECTO
String	PrivateAccess	changeable	

1.37.1.9 Atributo: sNivel

DESCRIPCIÓN	Es la descripción de los niveles de madurez de SW - CMM.
-------------	--

TIPO	VISIBILIDAD	PROPIEDAD	VALOR POR DEFECTO
String	PrivateAccess	changeable	

1.37.1.10 Atributo: sNombre

DESCRIPCION Esel nombre de SW - CMM.	
--	--

TIPO	VISIBILIDAD	PROPIEDAD	VALOR POR DEFECTO
String	PrivateAccess	changeable	

1.37.1.11 Atributo: sPracticaClave

DESCRIPCIÓN	Es la descripción de las prácticas claves de SW - CMM

TIPO	VISIBILIDAD	PROPIEDAD	VALOR POR DEFECTO
String	PrivateAccess	changeable	

1.37.1.12 Atributo: vtCaracteristicasComunes

DESCRIPCIÓN	Es una colección de características comunes.

TIPO	VISIBILIDAD	PROPIEDAD	VALOR POR DEFECTO
Vector	PrivateAccess	changeable	

1.37.2 OPERACIONES

1.37.2.1 Operación: bCargarServidorObjetoConsulta

DESCRIPCIÓN	TIPO RETORNO	VISIBILIDAD
Devuelve true si el servidor objetoConsulta está cargado, de lo contrario devuelve null.	boolean	PublicAccess

PROYECTO	KCMM VERSIÓN 2.0
SUBSISTEMA	GESTIÓN TUTOR
NOMBRE	1.38 Nivel
DESCRIPCIÓN	Es una abstracción de un nivel de madurez del modelo SW - CMM.
PREFIJO INSTANCIA	niv

1.38.1 ATRIBUTOS

1.38.1.1 Atributo: iEstadoLectura

DESCRIPCIÓN

TIPO	VISIBILIDAD	PROPIEDAD	VALOR POR DEFECTO
Int	PrivateAccess	changeable	

1.38.1.2 Atributo: ild

DESCRIPCIÓN Guarda el identificador de cada nivel de madurez.	
--	--

TIPO	VISIBILIDAD	PROPIEDAD	VALOR POR DEFECTO
Int	PrivateAccess	frozen	

1.38.1.3 Atributo: htAreasClaves

DESCRIPCION	Almacena una colección de áreas claves.
-------------	---

TIPO	VISIBILIDAD	PROPIEDAD	VALOR POR DEFECTO
Hashtable	PrivateAccess	changeable	

1.38.1.4 Atributo: proxyObjetoConsulta

DESCRIPCIÓN	Guarda una referencia del obieto consulta.

TIPO	VISIBILIDAD	PROPIEDAD	VALOR POR DEFECTO
IObjetoConsulta	PrivateAccess		

1.38.1.5 Atributo: sBeneficios

DESCRIPCIÓN	Guarda los beneficios que obtiene una empresa al alcanzar un nivel determinado.
	determinado.

TIPO	VISIBILIDAD	PROPIEDAD	VALOR POR DEFECTO
String	PrivateAccess	changeable	

1.38.1.6 Atributo: sDefinicion

DESCRIPCIÓN	Guarda la definición del nivel.

TIPO	VISIBILIDAD	PROPIEDAD	VALOR POR DEFECTO
String	PrivateAccess	changeable	

1.38.1.7 Atributo: sNombre

DESCRIPCION	Guarda el nombre del nivel.

TIPO	VISIBILIDAD	PROPIEDAD	VALOR POR DEFECTO
String	PrivateAccess	changeable	

1.38.1.8 Atributo: sTransicion

DESCRIPCIÓN	Describe los requisitos para alcanzar un nivel superior de acuerdo a la
DESCRIPCION	jerarquía planteada en SW - CMM

TIPO	VISIBILIDAD	PROPIEDAD	VALOR POR DEFECTO
String	PrivateAccess	changeable	

1.38.1.9 Atributo: sVisibilidad

DESCRIPCIÓN	Describe la capacidad de predicción que tiene la empresa cuando está en un
DESCRII CION	nivel determinado.

TIPO	VISIBILIDAD	PROPIEDAD	VALOR POR DEFECTO
String	PrivateAccess	changeable	

1.38.2 OPERACIONES

1.38.2.1 Operación: apcAdicionarAreaClave

DESCRIPCIÓN	TIPO RETORNO	VISIBILIDAD
Adiciona un área clave al nivel.		PublicAccess

1.38.2.2 Operación: apcDevolverAreaClave

DESCRIPCIÓN	TIPO RETORNO	VISIBILIDAD
Devuelve el área clave de un nivel.	AreaClave	PublicAccess

1.38.2.3 Operación: apcEliminarAreaClave

DESCRIPCIÓN	TIPO RETORNO	VISIBILIDAD
Elimina un área clave de un nivel.		PublicAccess

PROYECTO	KCMM VERSION 2.0
SUBSISTEMA	GESTIÓN TUTOR
VISIBILIDAD	PublicAccess
NOMBRE	1.39 ISwCmm
DESCRIPCIÓN	Declara todos los servicios que implementa la clase SwCmm.
PREFIJO INSTANCIA	

1.39.1 ATRIBUTOS

1.39.2 OPERACIONES

1.39.2.1 Operación: cargarSwCmm

DESCRIPCIÓN	TIPO RETORNO	VISIBILIDAD
Carga SW - CMM.		PublicAccess

1.39.2.2 Operación: ccDevolverCaracteristicaComun

DESCRIPCIÓN	TIPO RETORNO	VISIBILIDAD
Devuelve una característica común.	CaracteristicaComun	PublicAccess

1.39.2.3 Operación: iDevolverEstadoLectura

DESCRIPCIÓN	TIPO RETORNO	VISIBILIDAD
Devuelve el estado de la lectura de SW - CMM	int	PublicAccess

1.39.2.4 Operación: sDevolverAreaClave

DESCRIPCIÓN	TIPO RETORNO	VISIBILIDAD
Devuelve la descripción de un área clave.	String	PublicAccess

1.39.2.5 Operación: sDevolverArquitectura

DESCRIPCIÓN	TIPO RETORNO	VISIBILIDAD
Devuelve la descripción de la arquitectura de SW-CMM.	String	PublicAccess

1.39.2.6 Operación: sDevolverCaracteristicaComun

DESCRIPCIÓN	TIPO RETORNO	VISIBILIDAD
Devuelve la descripción de las características comunes de SW-CMM.	String	PublicAccess

1.39.2.7 Operación: sDevolverCaracteristicas

DESCRIPCIÓN	TIPORETORNO	VISIBILIDAD
Devuelve la descripción de las características de SW-CMM.	String	PublicAccess

1.39.2.8 Operación: sDevolverHistoria

DESCRIPCIÓN	TIPO RETORNO	VISIBILIDAD
Devuelve la descripción de la historia de SW-CMM.	String	PublicAccess

1.39.2.9 Operación: sDevolverNivel

DESCRIPCIÓN	TIPO RETORNO	VISIBILIDAD
Devuelve la descripción de los niveles de madurez de SW-CMM.	String	PublicAccess

1.39.2.10 Operación: sDevolverNombre

DESCRIPCIÓN	TIPO RETORNO	VISIBILIDAD
Devuelve el nombre de SW-CMM.	String	PublicAccess

1.39.2.11 Operación: vtDevolverTodasCaracteristicasComunes

DESCRIPCIÓN	TIPO RETORNO	VISIBILIDAD
Devuelve todas las características comunes de SW - CMM	Vector	PublicAccess

EVALUACIÓN

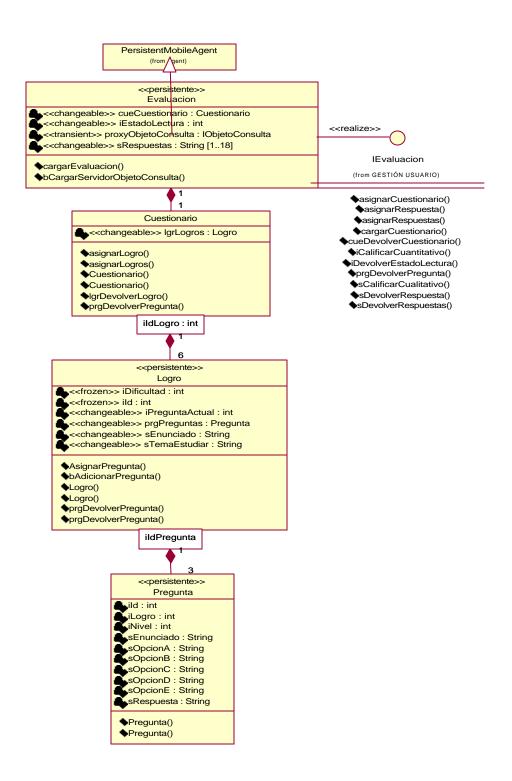


Diagrama de Clases 4 Evaluación

PROYECTO	KCMM VERSIÓN 2.0
SUBSISTEMA	GESTION TUTOR
VISIBILIDAD	PublicAccess
NOMBRE	1.40 Pregunta
	Un objeto pregunta tiene una interrogación y las posibles opciones que el usuario puede responde; tambien posee la opción correcta.
DESCRIPCIÓN	Todas las preguntas están asociadas, primero a un logro y también a un concepto para proponerlo de estudio en caso que la repuesta del usuario sea incorrecta.
PREFIJO INSTANCIA	prg

1.40.1 ATRIBUTOS

1.40.1.1 Atributo: ild

DESCRIPCIÓN	Es el identificador de la pregunta.

TIPO	VISIBILIDAD	PROPIEDAD	VALOR POR DEFECTO
Int	PrivateAccess		

1.40.1.2 Atributo: iLogro

DESCRIPCIÓN	Es el identificador del logro al cual está asociada la pregunta

TIPO	VISIBILIDAD	PROPIEDAD	VALOR POR DEFECTO
Int	PrivateAccess		

1.40.1.3 Atributo: iNivel

Guarda el identificador del nivel al que está asociada la pregunta.

TIPO	VISIBILIDAD	PROPIEDAD	VALOR POR DEFECTO
Int	PrivateAccess		

1.40.1.4 Atributo: sEnunciado

Es el enunciado de la pregunta.	DESCRIPCION	Es el enunciado de la pregunta.
---------------------------------	-------------	---------------------------------

TIPO	VISIBILIDAD	PROPIEDAD	VALOR POR DEFECTO
String	PrivateAccess		

1.40.1.5 Atributo: sOpcionA

DESCRIPCION	Es la opción A de la pregunta.		
-------------	--------------------------------	--	--

TIPO	VISIBILIDAD	PROPIEDAD	VALOR POR DEFECTO
String	PrivateAccess		

1.40.1.6 Atributo: sOpcionB

I ES la obción B de la predunta.	DESCRIPCIÓN	Es la opción B de la pregunta.
----------------------------------	-------------	--------------------------------

TIPO	VISIBILIDAD	PROPIEDAD	VALOR POR DEFECTO
String	PrivateAccess		

1.40.1.7 Atributo: sOpcionC

DESCRIPCION	Es la opción C de la pregunta.

TIPO	VISIBILIDAD	PROPIEDAD	VALOR POR DEFECTO
String	PrivateAccess		

1.40.1.8 Atributo: sOpcionD

DESCRIPCIÓN Es la o	oción D de la pregunta.
----------------------------	-------------------------

TIPO	VISIBILIDAD	PROPIEDAD	VALOR POR DEFECTO
String	PrivateAccess		

1.40.1.9 Atributo: sOpcionE

DESCRIPCIÓN	Es la opción E de la pregunta.

TIPO	VISIBILIDAD	PROPIEDAD	VALOR POR DEFECTO
String	PrivateAccess		

1.40.1.10 Atributo: sRespuesta

DESCRIPCION	Es la respuesta de la pregunta.
-------------	---------------------------------

TIPO	VISIBILIDAD	PROPIEDAD	VALOR POR DEFECTO
String	PrivateAccess		

1.40.2 OPERACIONES

1.40.2.1 Operación: Pregunta

DESCRIPCIÓN		TIPO RETORNO	VISIBILIDAD
	Constructor		PublicAccess

PROYECTO	KCMM VERSIÓN 2.0
SUBSISTEMA	GESTIÓN TUTOR
VISIBILIDAD PublicAccess	
NOMBRE	1.41 Logro
DESCRIPCIÓN	 -> Esta clase permite evaluar cualitativamente a los estudiantes, definiendo un logro como un conjunto de tres preguntas. El logro se cumple si las tres respuestas son correctas. -> Los logros tiene 3 grado de dificultad (bajo, medio y alto) el sistema debe escoger 2 logros por grado.
PREFIJO INSTANCIA	lgr

1.41.1 ATRIBUTOS

1.41.1.1 Atributo: iDificultad

DESCRIPCIÓN	Es el grado de dificultad del logro. Puede ser: 0 = Baja 1 = Media
	2 = Alta

TIPO	VISIBILIDAD	PROPIEDAD	VALOR POR DEFECTO
Int	PrivateAccess	frozen	

1.41.1.2 Atributo: ild

DESCRIPCIÓN	Es el identificador del logro.
-------------	--------------------------------

TIPO	VISIBILIDAD	PROPIEDAD	VALOR POR DEFECTO
Int	PrivateAccess	frozen	

1.41.1.3 Atributo: iPreguntaActual

DESCRIPCION	Es el identificador de la pregunta actual.

TIPO	VISIBILIDAD	PROPIEDAD	VALOR POR DEFECTO
Int	PrivateAccess	changeable	

1.41.1.4 Atributo: prgPreguntas

DESCRIPCIÓN	Es una colección de las preguntas asociadas al logro.

TIPO	VISIBILIDAD	PROPIEDAD	VALOR POR DEFECTO
Pregunta	PrivateAccess	changeable	

1.41.1.5 Atributo: sEnunciado

DESCRIPCION Es e	el enunciado del logro.
--------------------	-------------------------

TIPO	VISIBILIDAD	PROPIEDAD	VALOR POR DEFECTO
String	PrivateAccess	changeable	

1.41.1.6 Atributo: sTemaEstudiar

DESCRIPCION	Describe el tema a estudiar asociado al logro.

TIPO	VISIBILIDAD	PROPIEDAD	VALOR POR DEFECTO
String	PrivateAccess	changeable	

1.41.2 OPERACIONES

1.41.2.1 Operación: AsignarPregunta

DESCRIPCIÓN	TIPO RETORNO	VISIBILIDAD
Asigna una pregunta al logro.		PublicAccess

1.41.2.2 Operación: bAdicionarPregunta

DESCRIPCIÓN	TIPO RETORNO	VISIBILIDAD
Devuelve true si se pudo adicionar la pregunta, de lo contrario devuelve null.	Bolean	PublicAccess

1.41.2.3 Operación: Logro

DESCRIPCIÓN	TIPO RETORNO	VISIBILIDAD
Constructor de la clase logro.		PublicAccess

1.41.2.4 Operación: prgDevolverPregunta

DESCRIPCIÓN	TIPO RETORNO	VISIBILIDAD
Devuelve la pregunta que está asociada al logro, la	Pregunta	PublicAccess
cual se encuentra en la posición número pregunta.	i regunta	1 ubiloAccess

1.41.2.5 Operación: prgDevolverPregunta

DESCRIPCIÓN	TIPO RETORNO	VISIBILIDAD
Devuelve una pregunta asociada al logro.	Pregunta	PublicAccess

PROYECTO	KCMM VERSIÓN 2.0
SUBSISTEMA	GESTIÓN TUTOR
VISIBILIDAD	PublicAccess
NOMBRE	1.42 Cuestionario
DESCRIPCIÓN	-> Permite seleccionar aleatoriamente logros para realizar la evaluación> En cada cuestionario deben haber dos logros de cada grado de dificultad Total Logros por cuestionario: 6 - Total Preguntas por cuestionario : 18
PREFIJO INSTANCIA	cue

1.42.1 ATRIBUTOS

1.42.1.1 Atributo: IgrLogros

DESCRIPCION	Es un arreglo de logros.

TIPO	VISIBILIDAD	PROPIEDAD	VALOR POR DEFECTO
Logro	PrivateAccess	changeable	

1.42.2 OPERACIONES

1.42.2.1 Operación: asignarLogro

DESCRIPCIÓN	TIPO RETORNO	VISIBILIDAD
Asigna un logro al cuestionario.		PublicAccess

1.42.2.2 Operación: asignarLogros

DESCRIPCIÓN	TIPO RETORNO	VISIBILIDAD
Asigna los logros al cuestionario.		PublicAccess

1.42.2.3 Operación: Cuestionario

DESCRIPCIÓN	TIPO RETORNO	VISIBILIDAD
Constructor de la clase cuestionario.		PublicAccess

1.42.2.4 Operación: IgrDevolverLogro

DESCRIPCIÓN	TIPO RETORNO	VISIBILIDAD
Devuelve un logro del cuestionario.	Logro	PublicAccess

1.42.2.5 Operación: prgDevolverPregunta

DESCRIPCIÓN	TIPO RETORNO	VISIBILIDAD
Devuelve una pregunta asociada al logro.	Pregunta	PublicAccess

PROYECTO	KCMM VERSION 2.0
SUBSISTEMA	GESTIÓN TUTOR
VISIBILIDAD	PublicAccess
NOMBRE	1.43 Evaluacion
DESCRIPCIÓN	Además de abstraer el concepto de evaluación, esta clase implementa los servicios necesarios para poder realizar evaluaciones a los usuarios, identificar sus debilidades, fortalezas y temas a estudiar.
PREFIJO INSTANCIA	eva

1.43.1 ATRIBUTOS

1.43.1.1 Atributo: cueCuestionario

DESCRIPCIÓN	Guarda el cuestionario que será presentado por el estudiante.

TIPO	VISIBILIDAD	PROPIEDAD	VALOR POR DEFECTO
Cuestionario	PrivateAccess	changeable	

1.43.1.2 Atributo: iEstadoLectura

DESCRIPCION	Define el estado de lectura del cuestionario en la base de datos.
-------------	---

TIPO	VISIBILIDAD	PROPIEDAD	VALOR POR DEFECTO
Int	PrivateAccess	changeable	

1.43.1.3 Atributo: proxyObjetoConsulta

DESCRIPCIÓN	Guarda una referencia al obieto consulta.

TIPO	VISIBILIDAD	PROPIEDAD	VALOR POR DEFECTO
IObjetoConsulta	PrivateAccess	transient	

1.43.1.4 Atributo: sRespuestas

DESCRIPCIÓN	Guarda las respuestas digitadas por el usuario.
DESCRII CICIA	Las respuestas pueden ser: A, B, C, D, E.

TIPO	VISIBILIDAD	PROPIEDAD	VALOR POR DEFECTO
String [118]	PrivateAccess	changeable	

1.43.2 OPERACIONES

PROYECTO	KCMM VERSION 2.0
SUBSISTEMA	GESTIÓN TUTOR
VISIBILIDAD	PublicAccess
NOMBRE	1.44 IEvaluacion
DESCRIPCIÓN	Declara los servicios necesarios para poder realizar evaluaciones a los usuarios.
PREFIJO INSTANCIA	

1.44.1 ATRIBUTOS

1.44.2 OPERACIONES

1.44.2.1 Operación: asignarCuestionario

DESCRIPCIÓN	TIPO RETORNO	VISIBILIDAD
Asigna el cuestionario de la evaluación.		PublicAccess

1.44.2.2 Operación: asignarRespuesta

DESCRIPCIÓN	TIPO RETORNO	VISIBILIDAD
Asigna una respuesta.		PublicAccess

1.44.2.3 Operación: asignarRespuestas

DESCRIPCIÓN	TIPO RETORNO	VISIBILIDAD
Asigna todas las respuestas de un cuestionario.		PublicAccess

1.44.2.4 Operación: cargarCuestionario

DESCRIPCIÓN	TIPO RETORNO	VISIBILIDAD
Carga el cuestionario de la base de datos.		PublicAccess

1.44.2.5 Operación: cueDevolverCuestionario

DESCRIPCIÓN	TIPO RETORNO	VISIBILIDAD
Devuelve el cuestionario de evaluación.	Cuestionario	PublicAccess

1.44.2.6 Operación: iCalificarCuantitativo

DESCRIPCIÓN	TIPO RETORNO	VISIBILIDAD
-> Califica cuantitativamente. El puntaje asignado a cada pregunta correctamente contestada, es proporcional al grado de dificultad del logro al que está asociada, quedando de la siguiente forma: - Alto = 15 - Medio = 10 - Bajo = 5	Int	PublicAccess

1.44.2.7 Operación: iDevolverEstadoLectura

DESCRIPCION	TIPO RETORNO	VISIBILIDAD
Devuelve el estado de lectura del cuestionario.	Int	PublicAccess

1.44.2.8 Operación: prgDevolverPregunta

DESCRIPCIÓN	TIPO RETORNO	VISIBILIDAD
Devuelve una pregunta del cuestionario; devuelve la pregunta si es encontrada, de lo contrario devuelve null.	Pregunta	PublicAccess

1.44.2.9 Operación: sCalificarCualitativo

DESCRIPCIÓN	TIPO RETORNO	VISIBILIDAD
Califica cualitativamente identificando las	String	PublicAccess
debilidades, fortalezas y temas a estudiar.	String	1 abilicaccess

1.44.2.10 Operación: sDevolverRespuesta

DESCRIPCIÓN	TIPO RETORNO	VISIBILIDAD
Devuelve una respuesta del cuestionario.	String	PublicAccess

1.44.2.11 Operación: sDevolverRespuestas

DESCRIPCIÓN	TIPO RETORNO	VISIBILIDAD
Devuelve todas las respuestas del cuestionario.	String	PublicAccess

PROYECTO	KCMM VERSIÓN 2.0
SUBSISTEMA	GESTIÓN TUTOR
VISIBILIDAD	PublicAccess
NOMBRE	1.45 PersistentMobileAgent
DESCRIPCIÓN	Es la clase base que permite crear agentes que pueden moverse por si se necesita implementar el servicio de persistencia
PREFIJO INSTANCIA	

1.45.1 ATRIBUTOS

Ver documentación API de la plataforma para agentes móviles Grasshopper.

1.45.2 OPERACIONES

Ver documentación API de la plataforma para agentes móviles Grasshopper.

3 ANEXO C MANUAL DE CONCEPTOS BÁSICOS KCAGENTES

3.1 PREFACIO

3.1.1 Acerca de este Documento

Este documento define los conceptos de la tecnología de agentes móviles así como todos los conceptos que el usuario debe comprender para desenvolverse con propiedad en el ambiente KCAgentes. Es de aclarar que este no es un tutor de la tecnología de agentes móviles, la intención de introducir los conceptos de agentes móviles, es para dar al lector los conceptos fundamentales inherentes a ésta tecnología.

3.1.2 Documentos Relacionados

Este manual hace parte de toda la documentación que es entregada en el paquete de instalación. La documentación está compuesta por tres (3) manuales que permiten la total comprensión del sistema; estos manuales son:

- CONCEPTOS BÁSICOS: Explica los conceptos que el usuario debe comprender para hacer buen uso de la herramienta.
- MANUAL DE USUARIO: Describe los requerimientos y proceso de instalación de KCAgentes y además presenta una detallada descripción de los comandos y funcionalidades de la herramienta.
- MANUAL DEL PROGRAMADOR: Expone la manera como se realizan aplicaciones que pueden ser utilizadas en el ambiente KCAgentes.

Bienvenido a KCAgentes

"Tu herramienta amiga cuando deseas actualizar el software de tus clientes".

KCAgentes es una herramienta que permite a través de la plataforma de agentes Grasshopper 2.0, gestionar los agentes móviles que hacen parte de una aplicación que como empresa desarrolladora de software, decidiste distribuir a tus clientes. La gran utilidad de este software es que te facilita la actualización del software que tienes instalado en los computadores de tus clientes; sin la necesidad de trasladarte hasta donde se encuentran tus clientes o suspender las actividades que estén realizando tus clientes en el momento de la actualización.

Esta herramienta te permite actualizar el software de tus clientes en tiempo de ejecución sin afectar las tareas que ellos estén ejecutando en el momento de la actualización. El ambiente KCAgentes te permite obtener el estado del sistema en el momento justo antes de ejecutar la actualización, lo que te evita la pérdida de datos y el estado de los objetos que están siendo utilizados en el momento por tu cliente.

Ventajas

- ⇒ Te permite actualizar el software de tus clientes en tiempo de ejecución sin afectar las tareas que ellos estén ejecutando en el momento de la actualización.
- ⇒ Te permite obtener el estado del sistema en el momento justo antes de ejecutar la actualización, lo que te evita la pérdida de datos y el estado de los objetos que están siendo utilizados en el momento por tu cliente.
- ⇒ Te permite monitorear tus clientes, saber cuando ingresan al sistema, cuando salió, entre otras opciones.
- ⇒ Te permite mantener un control de todas las versiones de un producto que has distribuido a tus clientes.

3.2 CONCEPTOS

3.2.1 Mantenimiento o Actualización de Software

El mantenimiento de software constituye una fase de gran importancia dentro de la ingeniería de software. En el sentido que el software esta en constante cambio debido a la interacción con el cliente, quien en últimas define el éxito o fracaso del software en el mercado.

3.2.1.1 Clasificación de las Actualizaciones

3.2.1.1.1 Según su Causa

Los cambios que pueden generarse en el software obedecen a diferentes causas, de allí la naturaleza de los mismos; entre la causas se pueden destacar:

⇒ Actualización Correctiva

Busca solucionar errores encontrados por el cliente. Este tipo de mantenimiento además de pretender corregir los errores, lleva un alto ingrediente de manejo ejecutivo; debido a que están ligados generalmente a los términos contractuales que se establecieron con el cliente, y la buena imagen que la empresa desea ganar o mantener.

⇒ Actualización Adaptativa

Con el paso del tiempo es probable que las condiciones del entorno donde se implementó el software cambien (CPU, Sistema Operativo, reglas del negocio, entre otros). Por ello es necesario modificar el software para que se adapte al medio, generalmente el *Mantenimiento Adaptativo* se hace a petición del cliente.

⇒ Actualización Perfectiva

Una vez se empiece a utilizar el software, el cliente e incluso la misma empresa desarrolladora de software puede identificar funcionalidades adicionales que se van a reflejar en beneficios. El *Mantenimiento Perfectivo* lleva al software más allá de los requisitos funcionales originales.

⇒ Actualización Preventiva

Debido a los cambios y adelantos tecnológicos el software se va volviendo obsoleto, por eso es necesario que por medio del *Mantenimiento Preventivo* se cambie el software con el fin de que pueda corregirse, adaptarse o mejorarse de manera más fácil.

Cualquiera que sea la causa de cambios del software, las empresas desarrolladoras deben acudir a una estrategia para llevarlos a cabo de la manera más eficiente posible el mantenimiento o actualización.

3.2.1.1.2 Según el Recurso

Esta clasificación obedece al tipo de recurso que utilizarán para la actualización

⇒ Actualización Relacionada Con Archivos

Cuando las aplicaciones eran totalmente *monolíticas*, la única opción diferente a recompilar todo el proyecto, era adicionando, modificando o sustituyendo archivos con los que actuaba la aplicación, tales como: Bases de datos, bibliotecas dinámicas, archivos de inicio, archivos de configuración, ejecutables, entre otros. Esta técnica es aún utilizada.

⇒ Actualización Relacionada Con Componentes De Software

Con la aparición del *modelo de objetos distribuidos*, las actualizaciones pudieron hacerse de manera mucho más flexible. Para cambiar algunos aspectos de la lógica de aplicación o interfaces gráfica, sólo había que compilar de nuevo el objeto que se quisiera modificar, esto ocurrió gracias a tecnologías como *CORBA DCOM*, *RMI*, entre otras.

Más tarde gracias a la tecnología de *agentes móviles* se podía modificar un componente representado por un agente. La ventaja es significativa frente al modelo de objetos distribuidos ya que el agente puede recuperar sus estado; los agentes pueden moverse autónomamente y los usuarios no tiene la necesidad de reiniciar la aplicación.

⇒ Actualización Mixta

Esta actualización combina los dos tipos mencionados anteriormente.

3.2.1.1.3 Según Su Naturaleza

⇒ Actualización Automática

Es aquella donde la operaciones sobre los recursos utilizados (mover, eliminar, crear, reemplazar, etc); son realizadas por un componente de software sin la intervención de un usuario humano.

⇒ Actualización Manual

Es aquella donde la operaciones sobre los recursos utilizados (mover, eliminar, crear, reemplazar, etc); son realizadas por un usuario humano.

⇒ Actualización SemiAutomática

Es aquella donde las operaciones sobre los recursos utilizados (mover, eliminar, crear, reemplazar, etc); son realizadas por un componente de software; pero existe una intervención mínima de un usuario humano para decidir aspectos tales como: rutas de instalación, sistema operativo, etc.

3.2.2 Tecnología de Agentes Móviles

3.2.2.1 Agente móvil

Es un programa que puede viajar entre diferentes máquinas conservando su código, datos y estado de ejecución; con la autonomía de ejecutar, suspender o reanudar tareas, duplicarse o destruirse.

Grasshopper considera un agente como un programa que puede ejecutarse en un ambiente de agentes distribuido. Los agentes de Grasshopper pueden presentar los siguientes estados (Ver Figura 3):

Activo. Un agente presenta en estado activo al estar realizando una tarea. Cuando un agente se encuentra en este estado, sus métodos accesibles pueden ser invocados por otros componentes.

Suspendido. Un agente está suspendido si su tarea junto con su hilo de proceso son interrumpidos temporalmente. Cuando un agente se encuentra en este estado, sus métodos accesibles no pueden ser invocados por otros componentes, ni podrá comunicarse con otros agentes; pero puede en cualquier momento continuar la ejecución de sus tareas si pasa a estado activo.

Guardado. Este estado es controlado por el servicio de persistencia; cuando los agentes se encuentran en este estado su información relevante es guardada localmente y su instancia es retirada de la agencia. Los agentes que se encuentren en este estado pueden ser reactivados, es decir se crea una nueva instancia con los datos almacenados para que pueden continuar con la ejecución de sus tareas.

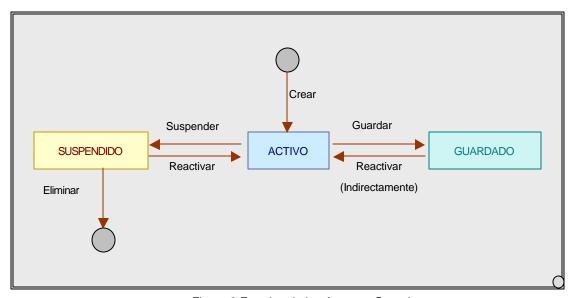


Figura 3 Estados de los Agentes Grasshopper

Existen dos tipos de agentes:

Agentes Móviles. Son agentes que se encuentran habilitados para moverse entre ubicaciones físicas de una red.

Agente Estacionario. Son agentes que no se encuentran habilitados para moverse entre ubicaciones físicas de red.

3.2.2.2 Ambiente de agentes distribuidos (Ver Figura 4)

Por sus siglas en inglés Distributed Agents Environment (DAE), es un conjunto de componentes que permiten realizar el ciclo de vida de los agentes móviles. Un DAE está compuesto por Regiones, Agencias, lugares y diferentes tipos de agentes.

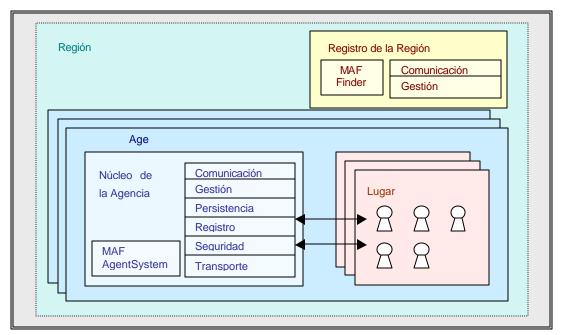


Figura 4 Arquitectura Grasshopper

3.2.2.3 Lugares

Los lugares son agrupaciones lógicas de agentes dentro de una agencia.

En el ambiente KCAgentes un lugar es una agrupación lógica de agentes, lo que constituye una funcionalidad.

3.2.2.4 Agencias

Una agencia es un ambiente de ejecución de agentes móviles y estacionarios, que proveen un conjunto de servicios que permiten realizar el ciclo de vida de los agentes. Las agencias están conformadas por uno o más lugares.

En KCAgentes el concepto de una agencia está estrechamente ligado a una aplicación; dado que precisamente la agencia es la que permite la ejecución de los agentes que conforman la aplicación.

- ⇒ **Núcleo de la Agencia.** Representa las funcionalidades mínimas requeridas para la ejecución de agentes, así pues una agencia soporta los siguientes servicios:
- ⇒ **Servicio de Comunicación.** Este servicio es de responsable de todas las interacciones remotas entre los componentes distribuidos de Grasshopper, por medio de CORBA, IIOP, java RMI o conexiones con SOCKET; estas dos últimas pueden ser protegidas por Secure Socket Layer (SSL). El servicio de comunicación soporta la comunicación síncrona, asíncrona, dinámica y multicast.
- ⇒ Servicio de Registro. Este servicio provee mecanismos para que las agencias identifiquen los lugares y agentes residentes en ellas, esto es importante para propósitos de administración externa, porque en cualquier momento se puede disponer de la información de las entidades registradas.

El nivel superior del servicio de registro es soportado por las regiones, ya que las agencias pueden conectarse a una región, así pues, ésta última puede administrar la información de todas las agencias conectadas y de todos los componentes de dichas agencias.

- ⇒ **Servicio de Gestión.** El servicio de gestión permite que un usuario externo pueda controlar y monitorear los agentes y lugares de una agencia. Entre otras las siguientes funcionalidades son soportadas:
 - * Crear, eliminar, suspender y ejecutar agentes y lugares.
 - * Obtener información específica de todos los lugares y agentes.
 - * Listar todos los agentes residentes en lugar en específico.
 - * Listar todos los lugares de una agencia.
 - * Además de las funcionalidades anteriores el servicio de administración permite que los usuarios humanos especifiquen las propiedades de comunicación y seguridad del sistema.

- ⇒ Servicio de Transporte. El servicio de transporte soporta la migración de agentes de una agencia a otra; permitiendo que el agente continúe su tarea en la agencia destino. Este servicio maneja la serialización de agentes y toda la coordinación de la migración, es realizada gracias al servicio de comunicación.
- ⇒ Servicio de Seguridad. Grasshopper soporta dos tipos de mecanismos de seguridad:
 - * Seguridad Externa. Protege la interacción remota entre los componentes distribuidos, permitiendo mejorar aspectos de autenticación, confidencialidad e integridad de los datos. Para este fin son usados los protocolos X.509 y SSL.
 - * Seguridad Interna. Restringe el acceso a los recursos de las agencias por parte de los agentes desautorizados, por medio de mecanismos de autenticación y verificación. Los servicios de seguridad interna están basados principalmente sobre los mecanismos de seguridad proveídos por JDK.
- ⇒ **Servicio de Persistencia.** El servicio de persistencia habilita el almacenamiento de agentes y lugares con el fin de recuperarlos cuando sea necesario. Existen dos tipos de persistencia:
 - * Persistencia Implícita. Cuando este servicio es activado, los lugares y agentes son automáticamente persistentes cuando son creados. Es de aclarar que la persistencia implícita es activada y configurada por los administradores de la agencia y no es visible para los agentes programadores.
 - * Persistencia Explícita. Existen tres mecanismos:

El primer mecanismo propone que los agentes sean guardados periódicamente, después de cierto intervalo de tiempo, sin suspender su tarea en ejecución.

El segundo mecanismo plantea que los agentes puedan solicitar su almacenamiento al servicio de persistencia, después de que las entidades no sean utilizadas durante un intervalo de tiempo. Los agentes permanecen registrados en la agencia y en el registro de la región, pudiendo ser reiniciados si otras entidades tratan de accederlo.

El tercer mecanismo es totalmente explícito ya que sugiere que los agentes sean guardados explícitamente por los usuarios humanos.

3.2.2.5 Región

El concepto de región facilita la administración de los componentes distribuidos en un ambiente Grasshopper. Tanto las agencias como los lugares pueden ser asociados con una región específica.

En el ambiente KCAgentes todas las aplicaciones que se encuentre deben estar registradas a una región, esto con el fin de que se puedan monitorear las aplicaciones, además de compartir servicios.

3.2.3 Ambiente para Mantenimiento de Software

Un ambiente para el mantenimiento de software (AMS) es un conjunto de herramientas y métodos que permiten el monitoreo, control y administración de componentes de software, en este caso agentes móviles; que fueron distribuidos (liberados) al cliente. El objetivo principal de un AMS, es que una vez se haya liberado un producto se puedan efectuar cambios reduciendo el impacto de los cambios al ente desarrollador.

No ha de confundirse el concepto Ambiente para el Mantenimiento de Software con el de *gestión de la configuración del software* que es una actividad de autoprotección que se realiza a lo largo del proceso de desarrollo de software y cuyo principal objetivo es identificar y controlar los cambios que ocurren durante el proceso. Su diferencia principal radica en el hecho que la gestión de configuración de software se realiza antes de liberar el producto, y el mantenimiento se realiza una vez se haya liberado el software.

Para comprender el funcionamiento de KCAgentes es necesario definir los siguientes conceptos:

3.2.3.1 Componente

Es la unidad básica en el ambiente de gestión de versiones. Un componente encapsula los datos necesarios para la creación de un agente dentro de una plataforma de agentes móviles.

Un componente tiene los siguientes atributos:

- ⇒ NOMBRE: Es el identificador del agente dentro de la aplicación, el nombre sirve para la localización de los agentes y sus servicios dentro de la aplicación.
- ⇒ DESCRIPCIÓN: Es importante para la persona encargada de realizar las actividades de mantenimiento, saber cual es la razón de ser de cada componente de software que constituye la aplicación; este campo permite poder describir el componente.
- ⇒ NOMBRE DE LA CLASE: Es un dato necesario para poder crear un agente móvil en cualquier plataforma de agentes, este campo contiene el nombre completo de la clase.
- ⇒ CODEBASE URL: Cualquier plataforma de agentes además de conocer el nombre de la clase, necesita saber donde está ubicada; por medio de este campo se puede definir el localizador uniforme de recurso (URL) que especifica la ubicación de la clase.

3.2.3.2 Funcionalidad

Dentro de una aplicación de agentes móviles es frecuente encontrar agentes que cooperan estrechamente para cumplir una tarea; una funcionalidad permite realizar agrupaciones lógicas de componentes. Generalmente los componentes que han de encontrarse en una misma funcionalidad tienen un objetivo común. Una funcionalidad tiene los siguientes datos:

- ⇒ *NOMBRE*: Es el identificador de la funcionalidad dentro de la aplicación, este nombre sirve para localizar los agentes que la constituyen.
- ⇒ DESCRIPCIÓN: Constituye una breve descripción de la misión de la funcionalidad dentro de la aplicación.

3.2.3.3 Versión

Es una agrupación lógica de funcionalidades, en esencia constituye el producto que está en disposición de ser distribuido a los clientes. En últimas una versión es una agrupación lógica de toda la información necesaria para generar y ejecutar una aplicación en una plataforma de agentes.

- ⇒ NOMBRE: Es el identificador de la aplicación.
- ⇒ DESCRIPCIÓN: Constituye una breve descripción de las funcionalidades que realiza la aplicación.

3.2.3.4 Proyecto KCAgentes

Los elementos que se mencionaron con anterioridad con frecuencia requieren ser editados eliminados, en fin, sobre esos elementos se pueden hacer diversas operaciones gracias al Módulo de Gestión de Versiones que hace parte del sistema KCAgentes

Para poder gestionar y guardar el estado de estos elementos, es necesario agruparlos en una entidad mucho mayor denominada Proyecto KCAgentes.

4 ANEXO D MANUAL DEL USUARIO KCAGENTES

4.1 PREFACIO

4.1.1 Acerca de este Documento

Este documento presenta aspectos relevantes para el uso de la herramienta KCAgentes; iniciando con la presentación de la herramienta, pasando por los aspectos referentes a la instalación para finalizar con los comandos y funcionalidades de KCAgentes.

4.1.2 Documentos Relacionados

Este manual hace parte de toda la documentación que es entregada en el paquete de instalación. La documentación está compuesta por tres (3) Manuales que permiten la total comprensión del sistema; estos manuales son:

- CONCEPTOS BÁSICOS: Explica los conceptos que el usuario debe comprender para hacer buen uso de la herramienta.
- MANUAL DE USUARIO: Describe los requerimientos y proceso de instalación de KCAgentes y además presenta una detallada descripción de los comandos y funcionalidades de la herramienta.
- MANUAL DEL PROGRAMADOR: Expone la manera como se realizan aplicaciones que pueden ser utilizadas en el ambiente KCAgentes.

Bienvenido a KCAgentes

"Tu herramienta amiga cuando deseas actualizar el software de tus clientes".

KCAgentes es una herramienta que permite a través de la plataforma de agentes Grasshopper 2.0, gestionar los agentes móviles que hacen parte de una aplicación que como empresa desarrolladora de software, decidiste distribuir a tus clientes. La gran utilidad de este software es que te facilita la actualización del software que tienes instalado en los computadores de tus clientes; sin la necesidad de trasladarte hasta donde se encuentran tus clientes o suspender las actividades que estén realizando tus clientes en el momento de la actualización.

Esta herramienta te permite actualizar el software de tus clientes en tiempo de ejecución sin afectar las tareas que ellos estén ejecutando en el momento de la actualización. El ambiente KCAgentes te permite obtener el estado del sistema en el momento justo antes de ejecutar la actualización, lo que te evita la pérdida de datos y el estado de los objetos que están siendo utilizados en el momento por tu cliente.

Ventajas

- ⇒ Te permite actualizar el software de tus clientes en tiempo de ejecución sin afectar las tareas que ellos estén ejecutando en el momento de la actualización.
- ⇒ Te permite obtener el estado del sistema en el momento justo antes de ejecutar la actualización, lo que te evita la pérdida de datos y el estado de los objetos que están siendo utilizados en el momento por tu cliente.
- ⇒ Te permite monitorear tus clientes, saber cuando ingresan al sistema, cuando salió, entre otras opciones.
- ⇒ Te permite mantener un control de todas las versiones de un producto que has distribuido a tus clientes.

IMPORTANTE: Es necesario que hayas leído el manual de CONCEPTOS BÁSICOS para poder comprender este documento.

4.2 INSTALACIÓN

4.2.1 Requerimientos

HARDWARE

Procesador: Procesador de 400 MHZ o superior

Ram: 256 MB

Video: Monitor SVGA con resolución de 800 X 600 píxeles

Tarjeta de Red

SOFTWARE:

Sistema Operativo: Tu sistema operativo debe admitir operaciones de red entre ellas proveer los servicios de Internet para crear sitios WEB, DNS, y DCHP entre otros. Los sistemas operativos para cuales te entregamos instaladores son Windows 2000 Server y Windows 2000 Advanced Server.

Máquina Virtual de Java: Debes tener instalada la Máquina Virtual de Java versión 1.3.1 – b4, si no la tienes instalada o tiene una versión anterior asegúrate de instalarla cuando ejecutes el instalador de este producto.

Grasshopper: Debes tener instalada la plataforma para agentes móviles Grasshopper versión 2.2.3 o superior, si no la tienes instalada o tiene una versión anterior asegúrate de instalarla cuando ejecutes el instalador de este producto.

4.2.2 Proceso de Instalación

4.2.2.1 Preparando la Instalación

⇒ Configurando el servidor

1. Establecer la dirección IP

Antes de establecer tu dirección IP debes estar seguro que æignarás una buena identificación de tu equipo en la red. Si ya la tienes establecida omite los siguientes pasos.

Para sistema Windows 2000 Server debes hacerlo de la siguiente manera:

- → Inicio
- → Configuración
- → Conexiones de red y de acceso telefónico (Ver Figura 5).

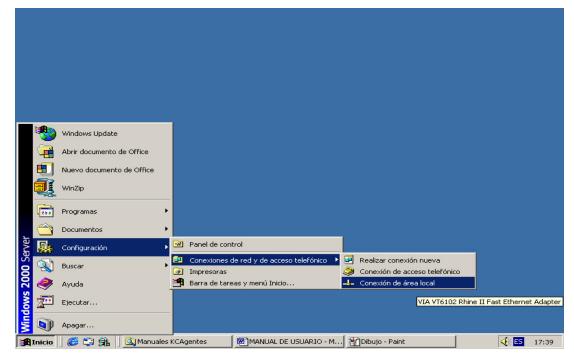


Figura 5 Conexiones de Red

Luego te aparecerá la caja de diálogo "Estado de Conexión de Área Local" (Ver Figura 6).



Figura 6 Estado de Conexión de área local

Haz click sobre el botón propiedades.



Figura 7 Propiedades de Conexión de área local

En la caja de diálogo propiedades (Ver Figura 7), ubícate en la pestaña *general* y busca en el listado *Protocolo TCP/IP*. Ubícate sobre él y haz click sobre propiedades.

En la caja de diálogo Propiedades de Protocolo de Internet (TCP/IP) (Ver Figura 8) puedes cambiar el valor de tu dirección IP y de tu DNS que en este caso debería tener la misma dirección IP.

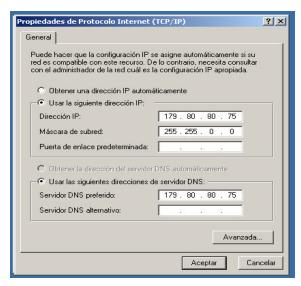


Figura 8 Propiedades de Protocolo Internet (TCP/IP)

Finalmente haz click en aceptar y reinicia el equipo para que los cambios surtan efecto.

Nota: Si tienes un sistema operativo diferente a WINDOWS 2000 Server consulta la ayuda para configurar tu dirección IP.

2. Configurar DNS y DCHP

Estos servicios son necesarios para asignar y resolver dinámicamente direcciones IP en tu red de área local; estos servicios además te permiten localizar todos los recursos disponibles en tu red. Consulta cómo configurar estos servicios en el manual de usuario o ayuda de tu sistema operativo.

3. Configurar Servidor Web

Este servicio te permite publicar recursos en tu red para que los diferentes computadores puedan tener acceso a ellos. Consulta en el manual de usuario o ayuda de tu sistema operativo como configurar este servicio.

4.2.2.2 Durante la Instalación

Después de haber configurado el servidor procedes a instalar KCAgentes en tu sistema operativo.

Antes de iniciar el proceso de instalación cerciórate que en el medio magnético donde te fueron distribuidos bs instaladores contenga las carpetas DATOS y SOPORTE, además de los archivo Instalar.bat y KCInstalador.class.

Ejecuta el archivo Instalar.Bat; y sigue las instrucciones del instalador.

4.2.2.3 Después de la instalación

En el directorio que instalaste KCAgentes se encuentran además de la aplicación, recursos que te serán de mucha utilidad (Ver Figura 9).



Figura 9 Contenido Directorio de Instalación

Verifica tu carpeta **Datos** (Ver Figura 10) y allí debes encontrar:

La carpeta **LIB** donde se encuentra la librería **KCAgentes.jar** que contiene todas las clases necesarias para crear aplicaciones basadas en la tecnología de agentes móviles y que puedan ser ejecutadas en el ambiente KCAgentes.

La carpeta **DOC** donde se distribuye la documentación del API de KCAgentes, allí se explican cada una de las clases que se encuentran en la librería, también encontrarás el Manual del Programador y el Manual de Conceptos Básicos.

En la carpeta **GRASSHOPPER** se encuentran las librerías y documentación de Grasshopper.

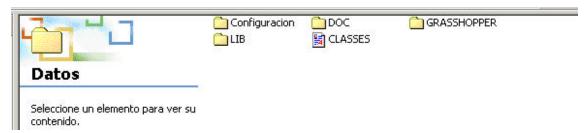


Figura 10 Contenido Carpeta Datos

En la carpeta Configuración (Ver Figura 11) encuentras:

- ⇒ La Carpeta **Base de Datos** donde deben ubicarse los archivos que deseas el gestor de persistencia pueda ver (Ver Manual del Programador).
- ⇒ La carpeta **Proyectos** donde se guardan los proyectos que desarrollas.
- ⇒ La carpeta **Usuarios** donde está la información de los usuarios administradores del sistema.



4.3 DURANTE LA EJECUCIÓN DE KCAgentes POR PRIMERA VEZ

Una vez hayas instalado KCAgentes tu sistema queda habilitado para que lo ejecutes; primero ejecuta KCMonitor.bat y luego KCAgentes.bat y puedes empezar a usar KCAgentes, esta es tu primera pantalla (Ver Figura 12).

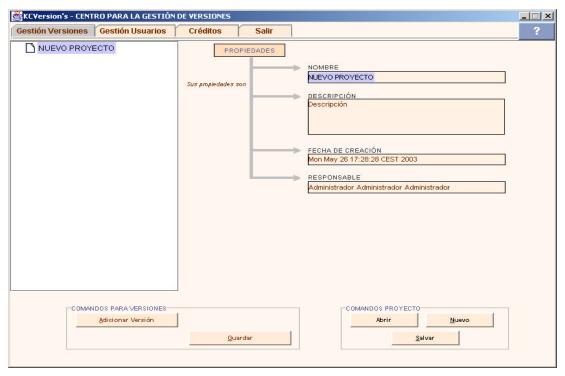


Figura 12 Ejecutado KCAgentes por primera vez

4.4 FUNCIONALIDADES Y COMANDOS

Antes de iniciar con cada uno de los comandos y funcionalidades es necesario que conozcas algunos términos inherentes a la interfaz gráfica de KCAgentes y que a partir de ahora se utilizarán para describir los comandos y funcionalidades (Ver Figura 13).

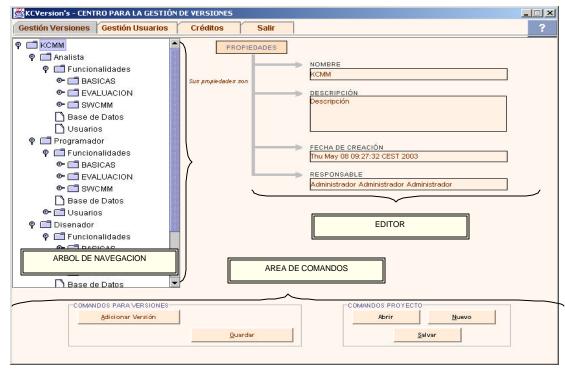


Figura 13 Elementos de la Interfaz Gráfica

Existe un área de navegación que se denomina **Árbol**, en esta sección puedes seleccionar cada uno de los elementos que hacen parte del ambiente KCAgentes.

En el área denominada *Editor*, puedes cambiar los valores del elemento que seleccionaste en el árbol.

Por último en la sección de *Comandos*, consigues los botones para ejecutar operaciones sobre los elementos seleccionados.

Para mayor comprensión del usuario, las funcionalidades serán presentadas por cada uno de los módulos del sistema.

4.4.1 Gestión Versiones (KCVersiones)

4.4.1.1 Crear Proyecto

Para crear un proyecto debes hacer click sobre el botón NUEVO (ALT + N), debes decidir si deseas guardar el proyecto que en dicho instante se encuentra abierto (Ver Figura 14).

Por defecto el proyecto se llama NUEVO PROYECTO, para cambiar los valores que están por defecto, debes ubicarte sobre el nombre del proyecto en el árbol y luego en el editor cambiar los datos del proyecto. Recuerda que los proyectos son guardados con extensión pry.

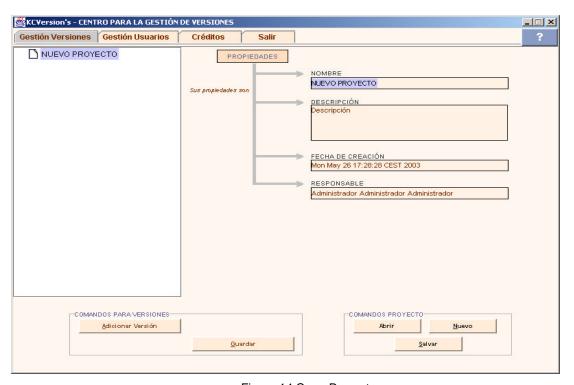


Figura 14 Crear Proyecto

4.4.1.2 Salvar Proyecto

Te permite guardar los cambios realizados al proyecto, haz click sobre el botón SALVAR (ALT + S) y el proyecto será guardado en la ruta: ...usr/Datos/Configuración/Proyectos (Ver Figura 15).

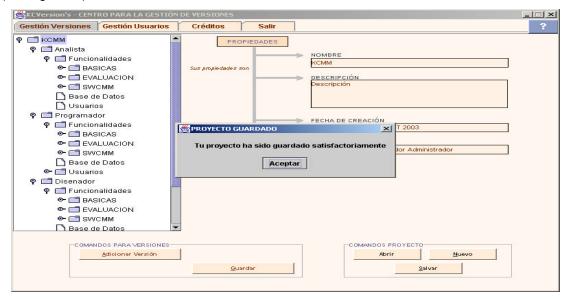


Figura 15 Salvar Proyecto

4.4.1.3 Abrir Proyecto

Para abrir un proyecto existente se debe hacer click sobre el botón *ABRIR (ALT + O)*. Debes decidir **si deseas guardar los cambios** al proyecto que en ese momento se encuentra abierto; posteriormente selecciona el proyecto que deseas abrir. (Ver Figura 16)

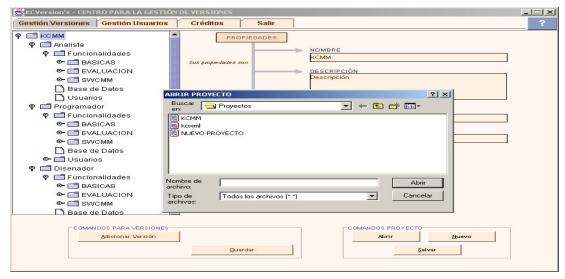


Figura 16 Abrir Proyecto

4.4.1.4 Adicionar Versión

Para adicionar una versión a un proyecto, se debe hacer click sobre el nombre del proyecto en el árbol e inmediatamente se activará el botón Adicionar Versión; luego se debe hacer click sobre el botón *ADICIONAR VERSIÓN (ALT + A)*, el sistema crea una nueva versión con los valores por defecto (Ver Figura 14).

4.4.1.5 Editar Versión

Para cambiar los valores de una versión, primero debes hacer click en el árbol sobre la versión que deseas editar, luego en el editor podrás cambiar los valores; **recuerda que para que los cambios surtan efecto debes hacer click sobre el botón** *GUARDAR (ALT + G)* (Ver Figura 17).

NOTA: Recuerda que no pueden quedar campos vacíos y al llenarlos ten en cuenta que existen caracteres no permitidos, asegúrate de no digitarlos:

- -> dos puntos :
 -> slash /
 -> back slash \
 -> comillas dobles "
 -> comillas sencillas '
- KCVersion's CENTRO PARA LA GESTIÓN DE VERSIONES _ | | × Gestión Versiones | Gestión Usuartos Créditos Salir ₱ III NUEVO PROYECTO P INJEVAVERBIÓN 1 ☐ Funcionalidades ☐ Base de Datos NOMBRE Usuarios [FECHA DE CREACIÓN RESPONSABLE Administracion Administration Administra AGESORES COMMUNICE PARA VERSIONE Rejublicar Agendia

Figura 17 Editar Versión

4.4.1.6 Eliminar Versión

Para eliminar una versión, debes ubicarte en el árbol sobre la versión que deseas eliminar y hacer click sobre el botón *ELIMINAR VERSIÓN (ALT + E)* (Ver Figura 17).

4.4.1.7 Editar Base de Datos de una Versión

Para editar la base de datos de una versión debes expandir en el árbol la versión ala cual deseas editarle la base de datos, posteriormente te ubicas sobre la etiqueta *Base de Datos*; te aparecerá una caja de diálogo que te permite cambiar los valores de la base de datos (Ver Figura 18).

NOTA: Recuerda que el DSN debe llevar la estructura jdbc:odbc:NOMBRE donde odbc es el nombre del controlador.

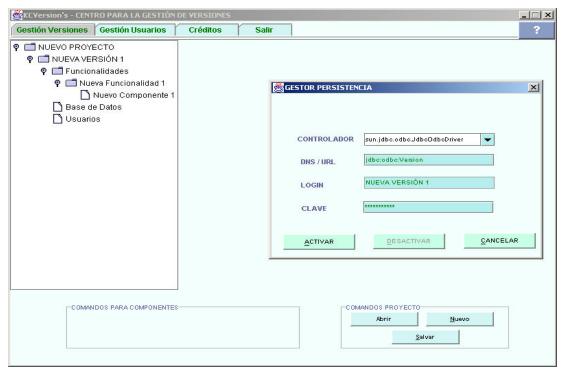


Figura 18 Editar Base de Datos

4.4.1.8 Adicionar Funcionalidad a una Versión.

Para adicionar una funcionalidad a una versión, debes expandir la versión sobre la cual deseas adicionar la funcionalidad, luego hacer click sobre la etiqueta *Funcionalidades* e inmediatamente se activará el botón *ADICIONAR FUNCIONALIDAD (ALT + A)* al cual debes hacerle click para que el sistema cree la nueva funcionalidad con los valores por defecto (Ver Figura 19).

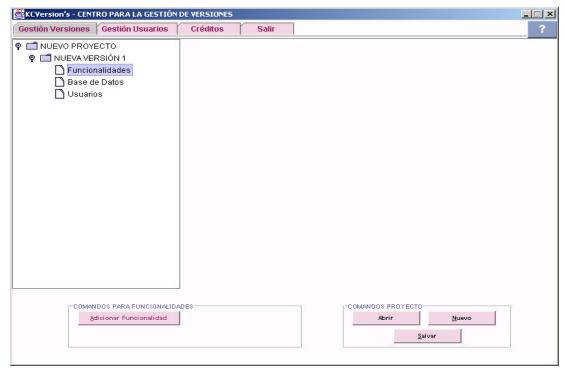


Figura 19 Adicionar Funcionalidad

4.4.1.9 Editar una Funcionalidad

Para Editar una funcionalidad debes ubicarte en el árbol sobre la funcionalidad que deseas editar y en el editor puedes cambiar sus valores; recuerda que para que los cambios surtan efecto debes hacer click sobre el botón GUARDAR (ALT + G) (Ver Figura 20)

NOTA: Recuerda que no pueden quedar campos vacíos y al llenarlos ten en cuenta que existen caracteres no permitidos, asegúrate de no digitarlos:

-> dos puntos :

slash /back slash \comillas dobles "comillas sencillas '

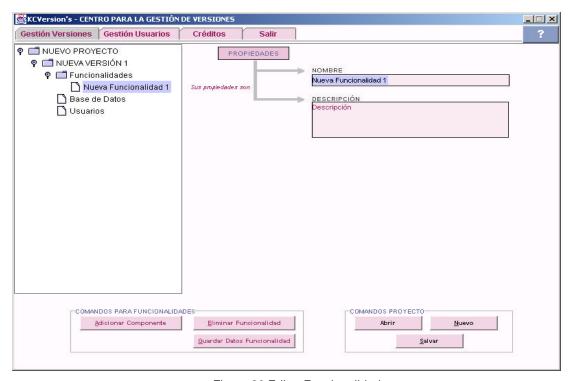


Figura 20 Editar Funcionalidad

4.4.1.10 Eliminar una Funcionalidad

Para eliminar una funcionalidad debes ubicarte en el árbol sobre la funcionalidad que deseas eliminar y hacer click sobre el botón *ELIMINAR FUNCIONALIDAD (ALT + E)* (Ver Figura 20).

4.4.1.11 Adicionar componente a una Funcionalidad

Para adicionar un componente a una funcionalidad, debes ubicarte en el árbol sobre la funcionalidad a la cual deseas adicionar un componente, luego debes hacer click sobre el botón *ADICIONAR COMPONENTE (ALT + A)*. El sistema creará un componente con los valores por defecto (Ver figura 20).

4.4.1.12 Editar un Componente

Para editar un componente debes ubicarte en el árbol sobre el componente que deseas editar y en el editor podrá cambiar los valores del componente; recuerda que para que los cambios surtan efecto debes hacer click sobre el botón GUARDAR (ALT + G) (Ver Figura 21)

NOTA: Recuerda que no pueden quedar campos vacíos y al llenarlos ten en cuenta que existen caracteres no permitidos, asegúrate de no digitarlos:

-> dos puntos :
-> slash /
-> back slash \
-> comillas dobles "
-> comillas sencillas '

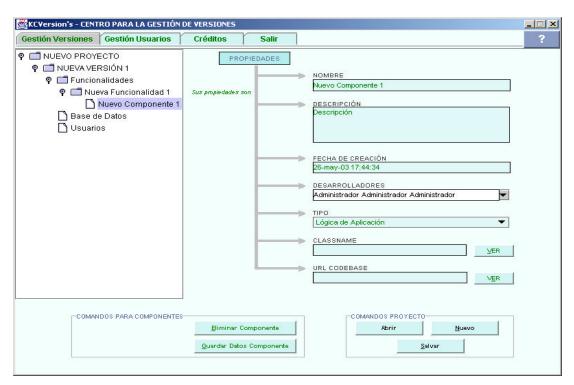
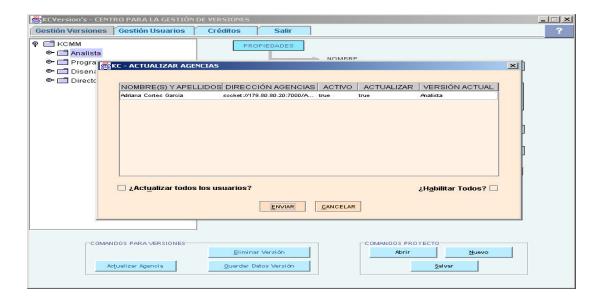


Figura 21 Editar Componente



4.4.2 GESTIÓN USUARIOS (KCUsuarios)

4.4.2.1 Adicionar Administrador

Para adicionar una administrador debes ubicarte en el árbol sobre la etiqueta ADMINISTRADORES y hacer click sobre el botón ADICIONAR (ALT + A) puedes editar los datos; pero recuerda que para que los cambios surtan efecto debes hacer click sobre el botón GUARDAR (ALT + G) (Ver Figura 23).

Nota: Esta funcionalidad solo está disponible para el administrador principal

Figura 23 Adicionar Administrador

4.4.2.2 Editar Administrador

Para cambiar los datos de un administrador debes ubicarte en el árbol sobre el administrador al cual deseas cambiarle los datos y en el editor puedes cambiar los datos; recuerda que para que los cambios surtan efecto debes hacer click sobre el botón *GUARDAR (ALT + G)* (Ver Figura 24)

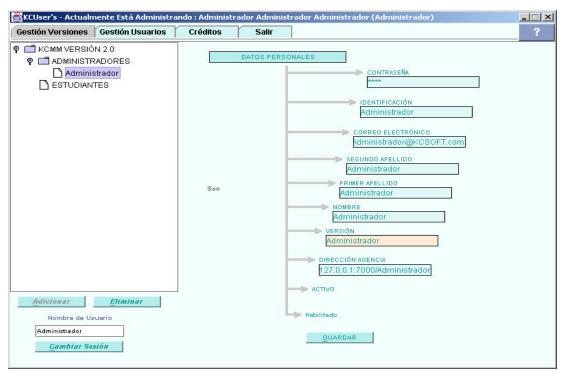


Figura 24 Editar Administrador

4.4.2.3 Eliminar Administrador

Para eliminar una administrador debes ubicarte en el árbol sobre el administrador que deseas eliminar y hacer click sobre el botón *ELIMINAR (ALT + E)* (Ver Figura 24)

Nota: Esta funcionalidad solo está disponible para el administrador principal, el cual nunca podrá ser eliminado

4.4.2.4 Cambiar Sesión

Para cambiar la sesión del administrador, debe ubicarte sobre el administrador al cual deseas cambiar sesión y hacer click sobre el botón *CAMBIAR SESIÓN (ALT + C)*, el sistema te pedirá la clave del nuevo administrador (Ver Figura 25).

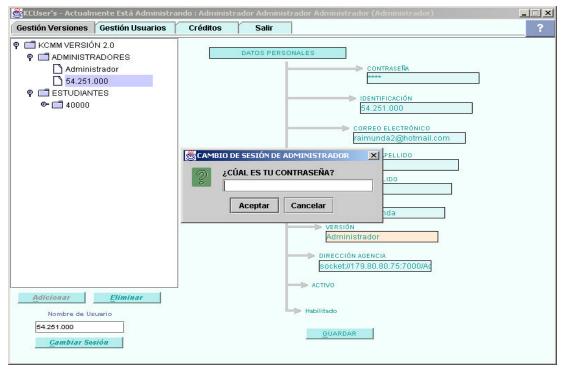


Figura 25 Cambiar Sesión

4.4.2.5 Monitorear Usuario

Para ver los datos de cualquier usuario registrado en el sistema te debes ubicar en el árbol sobre el usuario del cual deseas ver sus datos personales (Ver figura 26) y académicos (Ver figura 27).

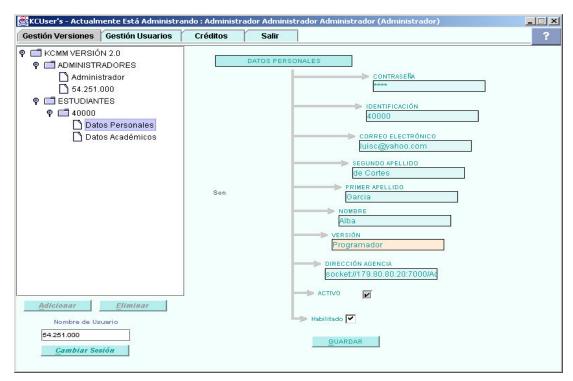


Figura 26 Ver Datos Personales Usuario

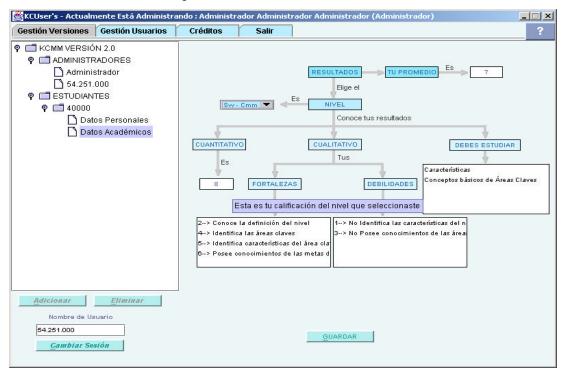


Figura 27 Datos Académicos Usuario

4.4.2.6 Habilitar y Deshabilitar un Usuario

Para realizar esta funcionalidad debes ubicarte en el árbol sobre el usuario que deseas habilitar o deshabilitar, luego ubícate en Datos Personales y posteriormente sobre el cuadro de chequeo "Habilitado" haz click según la operación que deseas hacer. Recuerda que los usuarios que estén deshabilitados no se le prestarán el servicio de actualización (Ver Figura 26).

5 ANEXO E MANUAL DEL PROGRAMADOR KCAGENTES

5.1 PREFACIO

5.1.1 Acerca de este Documento

Este documento define las consideraciones que un programador debe conocer para desarrollar aplicaciones que puedan actualizarse en el ambiente KCAgentes

En este documento se considera una aplicación como un conjunto de agentes móviles que desarrollan unas funcionalidades en común.

5.1.2 Documentos Relacionados

Este manual hace parte de toda la documentación que es entregada en el paquete de instalación. La documentación está compuesta por tres (3) Manuales que permiten la total comprensión del sistema; estos manuales son:

- CONCEPTOS BÁSICO: Explica los conceptos que el usuario debe comprender para hacer buen uso de la herramienta.
- MANUAL DE USUARIO: Describe los requerimientos y proceso de instalación de KCAgentes y además presenta una detallada descripción de los comandos y funcionalidades de la herramienta.
- MANUAL DEL PROGRAMADOR: Expone la manera como se realizan aplicaciones que pueden ser utilizadas en el ambiente KCAgentes.

Bienvenido a KCAgentes

"Tu herramienta amiga cuando deseas actualizar el software de tus clientes".

KCAgentes es una herramienta que permite a través de la plataforma de agentes Grasshopper 2.0, gestionar los agentes móviles que hacen parte de una aplicación que como empresa desarrolladora de software, decidiste distribuir a tus clientes. La gran utilidad de este software es que te facilita la actualización del software que tienes instalado en los computadores de tus clientes; sin la necesidad de trasladarte hasta donde se encuentran tus clientes o suspender las actividades que estén realizando tus clientes en el momento de la actualización.

Esta herramienta te permite actualizar el software de tus clientes en tiempo de ejecución sin afectar las tareas que ellos estén ejecutando en el momento de la actualización. El ambiente KCAgentes te permite obtener el estado del sistema en el momento justo antes de ejecutar la actualización, lo que te evita la pérdida de datos y el estado de los objetos que están siendo utilizados en el momento por tu cliente.

Ventajas

- ⇒ Te permite actualizar el software de tus clientes en tiempo de ejecución sin afectar las tareas que ellos estén ejecutando en el momento de la actualización.
- ⇒ Te permite obtener el estado del sistema en el momento justo antes de ejecutar la actualización, lo que te evita la pérdida de datos y el estado de los objetos que están siendo utilizados en el momento por tu cliente.
- ⇒ Te permite monitorear tus clientes, saber cuando ingresan al sistema, cuando salió, entre otras opciones.
- ⇒ Te permite mantener un control de todas las versiones de un producto que han distribuido a tus clientes.

IMPORTANTE: Es necesario que hayas leído los manuales de CONCEPTOS BÁSICOS y DEL USUARIO KCAGENTES, para poder comprender este documento.

5.2 ARQUITECTURA DE KCAGENTES

La arquitectura de KCAgentes (Ver Figura 28) está diseñada de tal manera que permite al administrador, ubicado una agencia administradora, gestionar todos los componentes de software que hacen parte de una aplicación, y a un cliente ubicado del lado del servidor, a acceder a los servicios de actualización de software que el administrador le provee.

KCAgentes provee algunos servicios que se describirán con mayor detalle en la próxima sección denominada Funcionamiento KCAgentes.

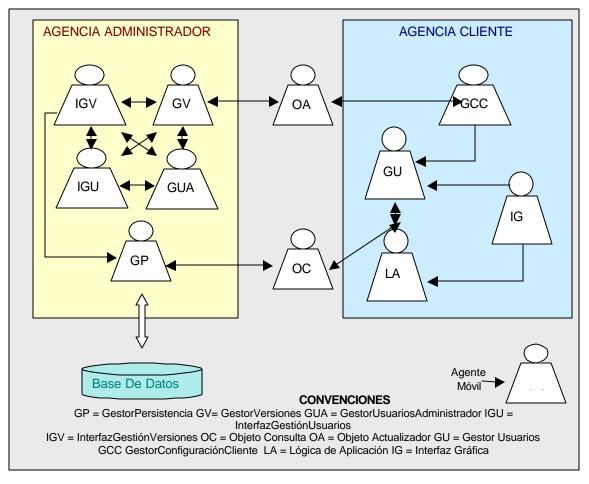


Figura 28 Arquitectura del ambiente KCAgentes

5.3 FUNCIONAMIENTO DE KCAgentes

Antes de explicar el funcionamiento del agente KCAgentes vale la pena destacar las siguientes consideraciones.

Los agentes que se encuentran en la agencia administrador se crean cuando se ejecuta la aplicación KCAgentes; estos agentes están siempre disponibles en el sistema; porque ofrecen servicios sin los cuales no puede funcionar KCAgentes (Ver figura 28)

Ahora entraremos en detalles en el funcionamiento de una aplicación en el ambiente KCAgentes:

Al lado del cliente, el usuario podrá interactuar con los agentes denotados como interfaz gráfica. Estos agentes a su vez según el diseño de la aplicación pueden solicitar algunas funcionalidades a los agentes catalogados como lógica de aplicación que implementan todas las reglas de negocio de la empresa.

En algunas aplicaciones es necesario que la lógica de aplicación interactúe o maneje algunos datos que se encuentra en dispositivos de almacenamiento secundarios.

Importante: Si en el diseño de la aplicación se definió que la información estará centralizada; los archivos y/o bases de datos deben estar ubicados en el computador donde se ejecuta la agencia administrador.

Cuando los agentes lógica de aplicación soliciten información centralizada deben seguir el siguiente proceso: El agente *Lógica de Aplicación* solicita información al *Objeto Consulta* a través de una Consulta SQL o una KCConsulta; luego el objeto consulta viaja hasta la agencia administrador e interactúa con el objeto *Gestor Persistencia*, Solicitando que realice la consulta; la consulta y la información del objeto que la solicitó son guardado en una cola de espera. El gestor persistencia abre el archivo o base de datos y ejecuta la consulta; el resultado de la consulta será entregado al objeto Consulta que a su vez lo entrega al objeto lógica de aplicación que lo solicitó.

5.3.1 Gestor Versiones

Este objeto es el encargado de realizar todas las tareas para la creación edición y eliminación de proyectos, versiones, funcionalidades y componentes; por medio de este objeto el objeto actualizador obtiene la información necesaria para crear una aplicación.

El objeto gestor versiones tiene un función denominada instalar versión que permite trasladar todos los objetos a una agencia destino. Este método es invocado cuando el gestor usuario crea un usuario.

5.3.2 Gestor Persistencia

El objeto gestor persistencia tiene la tarea de procesar directamente con la base de datos o los archivos; las peticiones recibidas por el objeto consulta esas peticiones pueden estar representadas por una consulta SQL o un consulta propia de KCAgentes denominada KCConsulta que se verá con mayor detalle más adelante.

El objeto gestor persistencia siempre está esperando una consulta por parte del objeto consulta, una vez la consulta es recibida, la ejecuta y recoge los datos de la consulta si la consulta fue de búsqueda, si la consulta fue de acción no hay necesidad de recoger datos; por último devuelve los resultados al objeto consulta. (Ver Figura 29)

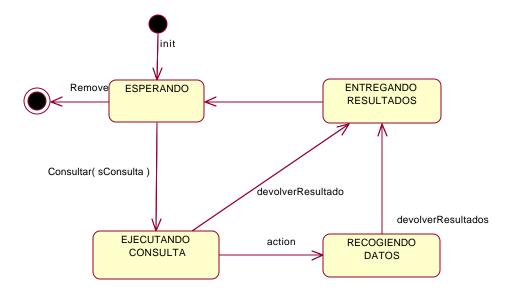


Figura 29 Diagrama de Estados Objeto Gestor Persistencia

5.3.2.1 Conexión a una Base de Datos

Para establecer una conexión con una base de datos en primera instancia hay que crear un *Origen De Datos* para ello consulta en la ayuda de tu sistema operativo. Ten en cuenta el nombre del origen de datos y los datos que se necesitaron para su configuración.

En el momento que creas una versión en el módulo administrador; puedes editar la información del gestor persistencia, haciendo click en el árbol sobre la etiqueta *Base de Datos*. La información solicitada corresponde a la que utilizaste para configurar el origen de datos.

El sistema provee al usuario de un listado de los principales controladores de bases de datos:

- ⇒ COM.ibm.bd2.app.DB2Driver
- ⇒ com.Informix.jdbc.ifxDriver
- ⇒ com.sybaaase.jdbc.SysDriver
- ⇒ interbase.interclient.Driver
- ⇒ oracle.jdbc.driver.OracleDriver
- ⇒ sun.jdbc.odbc.JdbcObdcDriver
- ⇒ webLogic.jdbc.Mssqserver4.Driver

5.3.2.2 Ejecución de Consultas

Cuando el objeto gestor persistencia está conectado a una base de datos, las ejecución de consultas es realizada gracias a una consulta SQL; sin embargo puede ocurrir que el desarrollador considere administrar la información por medio de archivos planos. Para guardar y recuperar la información de estos archivos la solicitud debe a través de una KCConsulta

La estructura de una Consulta es la siguiente COMANDO ARCHIVO CONTENIDO. Los COMANDOS pueden ser GUARDAR y LEER el argumento ARCHIVO es el nombre del archivo sobre el cual se desea hacer la operación y el argumento CONTENIDO será el contenido del archivo en el caso que se desee guardar, de lo contrario no se debe especificar el argumento.

IMPORTANTE: Los archivos a los que arriba se hace referencia deben estar ubicado en la carpeta ../DESTINO/Datos/Configuración/Base de Datos, donde DESTINO es la carpeta donde se instaló KCAgentes.

5.3.3 Objeto Actualizador

Este objeto es el encargado de viajar hasta la agencia cliente y a partir de la información proporcionada por el gestor de versiones invocar servicios a la plataforma de agentes para la eliminación y adición de componentes. El método que permite que el objeto realice una actualización es Actualizar() (Ver Figura 30).

Cada vez que el objeto actualizador ejecuta una actualización guarda un log con todas las acciones ejecutadas en /DESTINO/Datos/Configuración/Usuario/ADMINISTRADOR, donde DESTINO es la carpeta donde instaló KCAgentes y ADMINISTRADOR es la carpeta del administrador que ejecutó la actualización.

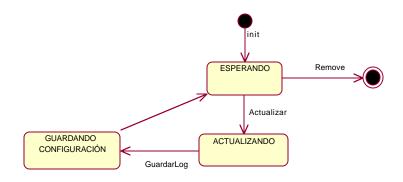


Figura 30 Diagrama de Estados Objeto Actualizador

5.3.4 Objeto Consulta

El programador debe tener en cuenta que para desarrollar cualquier aplicación que necesite recuperar datos que se encuentran centralizados, necesita un Objeto Consulta que es el encargado de recibir las peticiones de los agentes lógica de aplicación; viajar hasta la agencia administrador para solicitarle al Gestor Persistencia que procese la consulta y luego serializar los datos y llevarlo de vuelta al objeto lógica de aplicación que originalmente hizo la petición. (Ver Figura 31)

La existencia de un objeto consulta en una aplicación es necesaria para su funcionamiento debido a que el objeto Gestor Persistencia (Ver numeral 5.3) solicita constantemente sus servicios.

Puedes crear tu propio objeto consulta que herede de la clase kcagentes.LogicaAplicacion.GestionSistema.ObjetoConsulta Es necesario que implementes el método abstracto asignarNombreVersion. En este método hay que asignar a la variable sNombreVersion el nombre de la versión a la que pertenece el objeto.

Un objeto consulta está esperando constantemente que le soliciten una consulta; en el momento que cualquier objeto de la aplicación le realiza una consulta, el objeto viaja al nodo servidor para ejecutar la consulta, posteriormente recoge los datos tras realizar la consulta, viaja al nodo cliente y entrega los resultados al objeto que los solicitó.

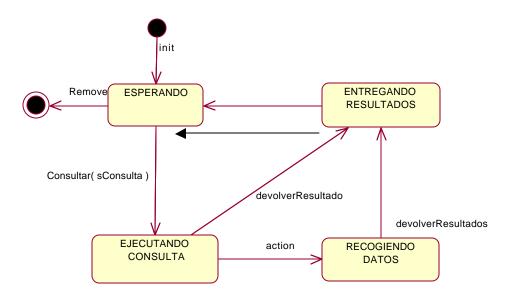


Figura 31 Diagrama de Estados Objeto Consulta

IMPORTANTE: Para que la aplicación funcione debes cerciorar que el nombre que asignaste en el método asignarNombreVersion del objeto Consulta sea el mismo que se especificará cuando se cree una versión en el módulo gestión versiones

EJEMPLO



Figura 32 Edición Gestor Persistencia en el Módulo de Gestión Versiones

Si en el módulo Gestión Versiones adicionaste una versión con el nombre Nueva Version 1(Ver Figura 32) la línea de código en el método æignarNombreVersion debió haber quedado:

public class MiObjetoConsulta extends ObjetoConsulta

import kcagentes.LogicaAplicacion.GestionSistema.ObjetoConsulta;

{
 public void asignarNombreVersion()
 {
 this.sNombreVersion = "Nueva Version 1";
 }
}

5.3.5 Gestor Usuarios

El objeto gestor usuarios es de gran importancia para el desarrollo de una aplicación que trabaje en el ambiente KCAgentes; la función de mayor importancia para este objeto, es la de registro del usuario en el ambiente; sin ello el usuario final no podría acceder a las

funcionalidades compartidas. Del lado del servidor guarda la información de todos los usuarios del sistema.

Puedes crear tu propio objeto Gestor Usuarios haciendo una clase que herede de KCAgentes.LogicaAplicacion.GestionUsuarios.GestorUsuarios EJEMPLO

import kcgentes.LogicaAplicacion.GestionUsuario.GestorUsuarios;
public class MiGestorUsuario extends GestorUsuarios
{ }

5.3.6 Interfaz Gestión Usuarios e Interfaz Gestión Versiones

Estas interfaces gráficas le permiten al administrador realizar las tareas comunes inherentes a un sistema de mantenimiento de software, por medio de comandos sencillos.

5.3.7 Gestor Configuración Cliente

Este agente se encuentra en el lado del cliente es el encargado de mantener la información referente a la configuración de la aplicación cliente; esta información es importante en el momento de guardar y recuperar una aplicación en el servidor.

Puedes crear tu propio objeto Gestor Configuración Cliente haciendo una clase que herede de

kcagentes.LogicaAplicacion.GestionSistema.GestorConfiguracionCliente.

EJEMPLO

import kcagentes.LogicaAplicacion.GestionSistema.GestorConfiguracionCliente
public class MiGestorConfiguracionCliente extends GestorConfiguracionCliente
{ }

IMPORTANTE: Si deseas guardar una aplicación en el servidor, debes invocar el método action() del objeto GestorConfiguracionCliente.

5.3.8 Lógica Aplicación e Interfaz Gráfica

Estos agentes hacen referencia a aquellos objetos que el desarrollador codificó y constituyen en esencia la aplicación que se distribuirá a los clientes.

Para que estos agentes sean aceptados en el sistema deben cumplir las siguientes condiciones:

Los agentes admitidos dentro del ambiente KCAgentes en primera instancia deben ser agentes desarrollados con el API de Grasshopper y heredar de las clases *MobileAgent, PersistentMobileAgent, StationaryAgent o PesistentStationaryAgent*; además de haber implementado la interfaz *IKcAgente*

5.3.8.1 La Interfaz IKCAgente

Esta interfaz define los métodos para guardar y recuperar el estado de los agentes; los agentes deben implementar esta interfaz para que sean admitidos en el ambiente KCAgentes

La implementación de los métodos definidos por **IKCAgente** difieren por cada agente, es decir, el programador debe definir cuales serán los atributos que desea guardar; cual será el formato para guardar esos atributos y donde guardará esos atributos.

Los métodos definidos por la interfaz **IKCAgentes** son:

bSerializar() Este método guarda el estado de un agente en un dispositivo de almacenamiento secundario.; devuelve true si el estado del agente pudo ser guardado, de lo contrario devuelve false.

sDesSerializar: Este método recupera el estado de un agente ; devuelve null si el estado pudo ser recuperado, de lo contrario devuelve un String.

IMPORTANTE: Para que los agentes automáticamente puedan recuperar y guardar su estado; El método bSerializar debe ser invocado desde el método beforeRemove de Grasshopper y el método sDesSerializar debe ser invocado en el método Init de Grasshopper.

6 ANEXO F MANUAL DEL USUARIO TUTOR KCMM

6.1 PREFACIO

6.1.1 Acerca de este Documento

Este documento presenta aspectos relevantes para el uso de la herramienta KCMM;

iniciando con la presentación de la herramienta, pasando por los aspectos referentes a la

instalación para finalizar con los comandos y funcionalidades.

Bienvenido a KCMM "La manera fácil de aprender

SW – CMM (modelo de madurez de capacidad para software)

KCMM es un tutor de SW CMM, que utiliza mapas conceptuales como organizador del

contenido temático y presenta una interfaz personalizada de acuerdo a los principales perfiles

en los proceso de desarrollo de software (Analista, Diseñador, Programador, Director de

Proyectos).

Además de permitir navegar por los contenidos de SW CMM, el tutor tiene un módulo

de evaluación que permite al usuario evaluar lo aprendido y partir de ello generar los resultados

que permiten identificar las debilidades y fortalezas del estudiante.

El tutor KCMM por estar desarrollado bajo la perspectiva de agentes móviles, permite

ser actualizado por medio de la herramienta KCAgentes.

6.2 INSTALACIÓN

6.2.1 Requerimientos

HARDWARE

Procesador: Procesador de 400 MHZ o superior

Ram: 256 MB

Video: Monitor SVGA con resolución de 800 X 600 píxeles

Tarjeta de Red

170

SOFTWARE:

Sistema Operativo: Sistema operativos Windows.

Máquina Virtual de Java: Debes tener instalada la Máquina Virtual de Java versión 1.3.1 – b4, si no la tienes instalada o tiene una versión anterior asegúrate de instalarla cuando ejecutes el instalador de este producto.

Grasshopper: Debes tener instalada la plataforma para agentes móviles Grasshopper versión 2.2.3 o superior.

6.2.2 Proceso de Instalación

6.2.2.1 Preparando la Instalación

Antes de iniciar el proceso de instalación debes cerciorarte de haber establecido tu dirección IP y haber configurado tu servidor DNS que te permita resolver la dirección del servidor de agentes. Para configurar el servidor DNS realiza los siguientes pasos:

Abre INICIO - PANEL DE CONTROL (Ver Figura 33).

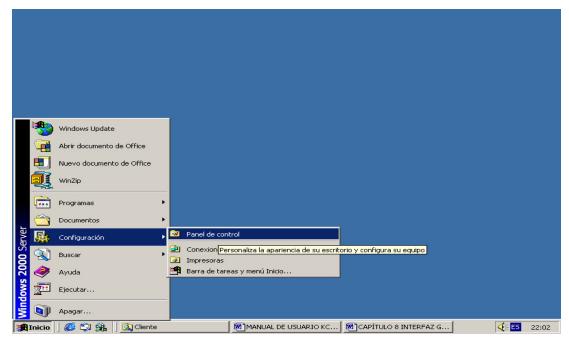


Figura 33 Inicio - Panel de Control

Estando en el panel de control abre el ícono Red (Ver Figura 34).

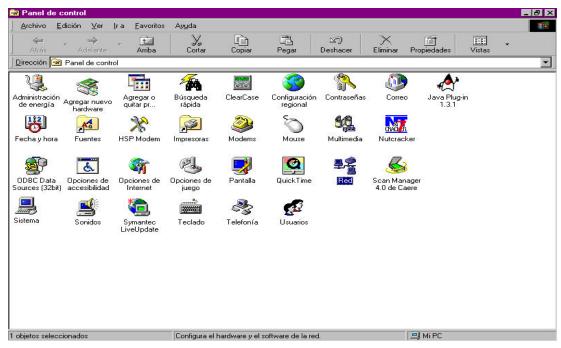


Figura 34 Panel de Control

Luego en la caja de dialogo Red selecciona el protocolo TCP / IP de asociado a la tarjeta TCP / IP que deseas configurar y haz click sobre el botón *Propiedades* (Ver Figura 35).

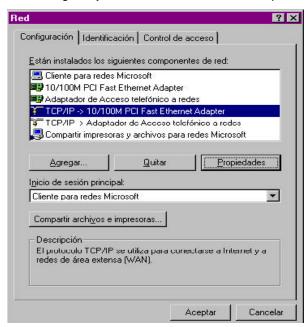


Figura 35 Red

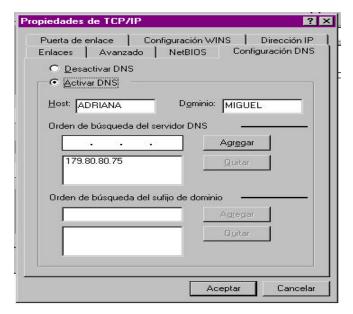


Figura 36 Propiedades TCP / IP

Para finalizar haz click en aceptar y reinicia el equipo para que los cambios surtan efecto

6.2.2.2 Durante la Instalación

Después de haber configurado el DNS procedes a instalar KCMM en tu Sistema Operativo. Ejecuta el archivo Instalar.Bat; y sigues las instrucciones del instalador.

6.2.2.3 Después de la Instalación

En el directorio que instalaste KCAgentes se encuentran además de la aplicación, los archivos KCClienteA y KCClienteB y la carperta **DATOS** (Ver Figura 37).



Figura 37 Contenido Directorio de Instalación

Verifica tu carpeta **Datos** y allí debes encontrar el archivo classes.jar y las carpetas **DOC**, donde se encuentra este manual de usuario; la Carpeta **GRASSHOPPER** donde se encuentran las librerías y documentación de Grasshopper; y la carpeta **CONFIGURACIÓN** (Ver Figura 38).

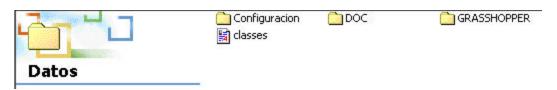


Figura 38 Contenido Carpeta Datos

6.3 DURANTE LA EJECUCIÓN DE KCMM POR PRIMERA VEZ

Una vez haya instalado KCAgentes tu sistema queda habilitado para que lo ejecutes, primero ejecuta KCMM.bat y puedes empezar a usar KCMM. Lo primero que debes hacer es registrarte al sistema. (Ver Figura 39)

6.4 COMPONENTES DE LA INTERFAZ

Es importante que sepas que la interfaz gráfica de KCMM está compuesta por cinco (5) elementos:

- ⇒ Mapa conceptual del tema
- ⇒ Control de navegación (Difiere de acuerdo al perfil del usuario)
- ⇒ Consola para visualizar la información.
- ⇒ Ayuda.
- ⇒ Elementos lúdicos.

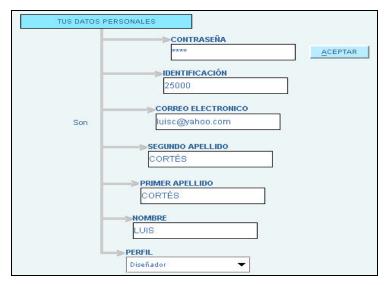


Figura 39 Ejecutando KCMM por primera vez

6.4.1 Mapa conceptual del tema o funcionalidad

Para el tutor KCMM se realizaron diez (10) mapas conceptuales que permiten visualizar:

- ⇒ Registro de usuario. (Ver Figura 39)
- ⇒ Información referente a SW CMM. (Ver Figura 40)

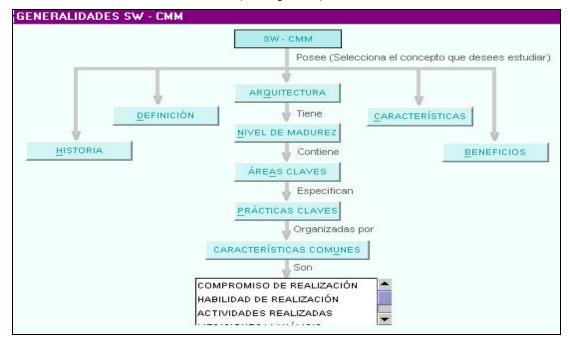


Figura 40 Mapa Conceptual: Conceptos Generales Sw - Cmm

⇒ Información referente a los niveles de SW - CMM. (Ver Figura 41)

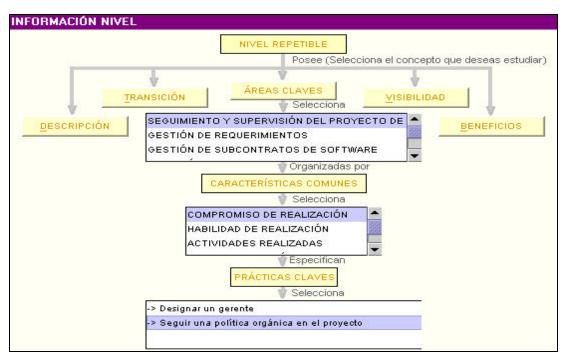


Figura 41 Mapa Conceptual: Niveles de Sw - Cmm

⇒ Evaluación correspondiente a cada nivel estudiado. (Ver Figura 42)

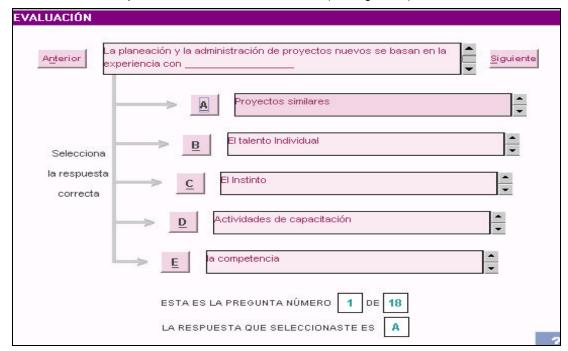


Figura 42 Mapa Conceptual: Evaluación

⇒ Los resultados de las evaluaciones presentadas por el usuario. (Ver Figura 43)

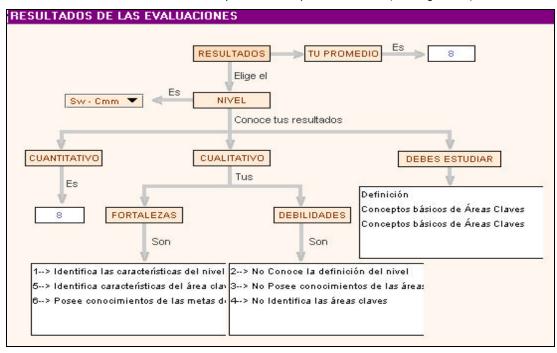


Figura 43 Mapa Conceptual: Resultados Evaluación

⇒ Los créditos del grupo desarrollador del proyecto. (Ver Figura 44)

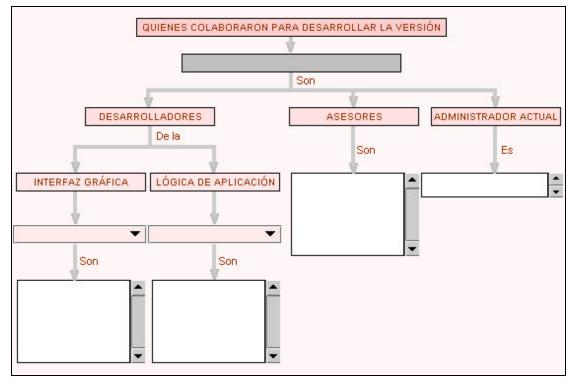


Figura 44 Mapa Conceptual: Créditos

6.4.2 Controles para la Navegación

Este componente fue implementado de forma diferente para cada perfil de usuario, buscando que los ambientes creados sean diferentes. A continuación se describen:

6.4.2.1 Control para la Navegación en la Interfaz Gráfica Analista

El control utilizado es el menú el cual se encuentra en la parte superior izquierda de las diferentes interfaces gráficas y se distingue con el nombre "DESEAS IR A" (Ver Figura 45); al desplegarse muestra los diferentes destinos que el Tutor proporciona al usuario.

Inicialmente el usuario tiene acceso a las interfaces gráficas de registro, Conceptos Generales, Evaluación y Créditos; a medida que presenta y aprueba las evaluaciones se le van activando los niveles superiores.

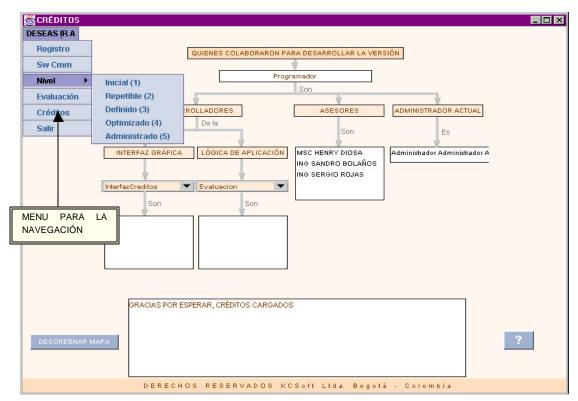


Figura 45 Control para la Navegación en la Interfaz Gráfica del Perfil Analista

6.4.2.2 Control para la Navegación en la Interfaz Gráfica Director de Proyectos

El control utilizado es el combo el cual se encuentra en la parte inferior derecha de las diferentes interfaces gráficas y se distingue con el nombre "DESEAS IR A" (Ver Figura 46); al desplegarse muestra los diferentes destinos que el Tutor proporciona al usuario.

Inicialmente el usuario tiene acceso a las interfaces gráficas de registro, Conceptos Generales, Evaluación y Créditos; a medida que presenta y aprueba las evaluaciones se le van activando los niveles superiores.

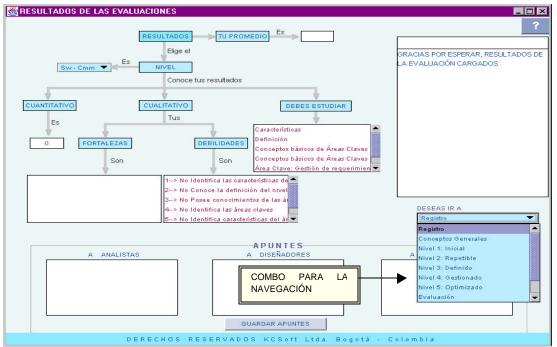


Figura 46 Control para la Navegación en la Interfaz Gráfica del Perfil Director De Proyectos

6.4.2.3 Menú de Navegación Interfaz Gráfica Diseñador

El control utilizado es el árbol el cual se encuentra en la parte superior izquierda de las diferentes interfaces (Ver Figura 47); donde cada una de sus hojas permite acceder a los diferentes destinos que el Tutor proporciona al usuario.

Inicialmente el usuario tiene acceso a las interfaces gráficas de registro, Conceptos Generales, Evaluación y Créditos; a medida que presenta y aprueba las evaluaciones se le van activando los niveles superiores.

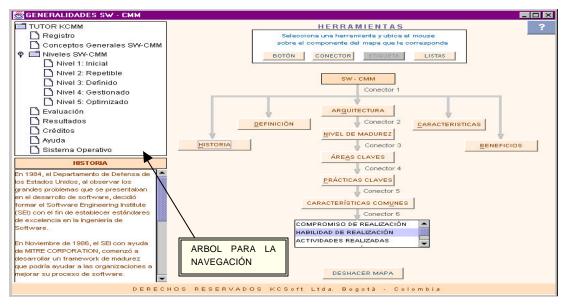


Figura 47 Control para la Navegación en la Interfaz Gráfica del Perfil Diseñador

6.4.2.4 Menú de Navegación Interfaz Programador

Los controles utilizados son las pestañas las cuales se encuentran en la parte inferior de las diferentes interfaces gráficas. (Ver Figura 48)

Inicialmente el usuario tiene acceso a las interfaces gráficas de registro, Conceptos Generales, Evaluación y Créditos; a medida que presenta y aprueba las evaluaciones se le van activando los niveles superiores.

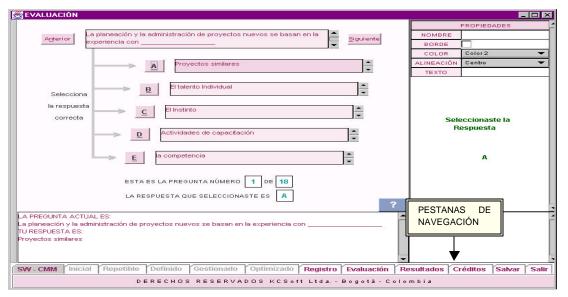


Figura 48 Control para la Navegación en la Interfaz Gráfica del Perfil Programador

6.4.3 Consola para Visualizar la Información de los Diferentes Temas.

Todas las interfaces gráficas independientemente del perfil al que estén asociadas poseen una consola (Ver Figura 49) en la que se visualiza:

- La información del Modelo SW CMM
- Los diferentes mensajes y errores que puedan surgir en la ejecución de la aplicación.
- Las acciones que el usuario debe realizar en algunas interfaces gráficas.

Su ubicación, color y tamaño depende del perfil al que esté asociada.

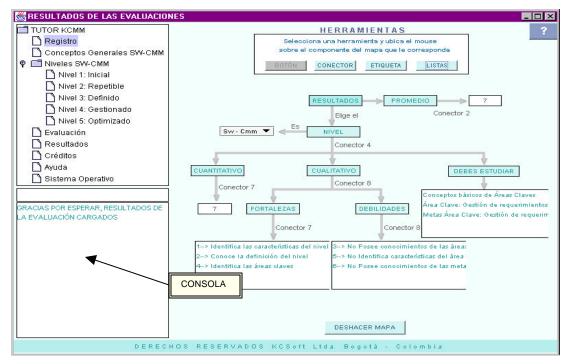


Figura 49 Consola de Información y Mensajes

6.4.4 Ayuda

Para visualizar la ayuda se debe hacer click sobre su ícono (Ver Figura 50), a continuación desplazarse al componente del que se desea recibir ayuda y hacer click sobre éste; inmediatamente se visualizará la ayuda correspondiente. (Ver Figura 51)



Figura 50 Ícono de Ayuda al Usuario



Figura 51 Visualización de Ayuda al Usuario

6.4.5 Elemento Lúdico

6.4.5.1 Elemento Lúdico para la Interfaz Gráfica del Perfil Analista

Al analista le atrae poder organizar sistemáticamente la realidad para obtener representaciones abstractas que le permitan visualizar, conocer o actuar sobre esa realidad.

Teniendo en cuenta lo anterior, la interfaz ha sido diseñada en forma de rompecabezas, compuesto por 16 fichas que en su conjunto forman cada mapa conceptual. Para que el analista acceda a las diferentes funcionalidades del tutor debe ordenarlo. (Ver Figura 52)

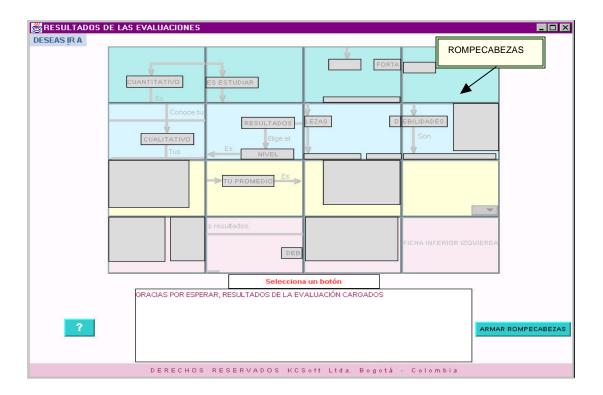


Figura 52 Elemento Lúdico (Rompecabezas) de la Interfaz Grafica del Perfil Analista

Este puzzle puede desorganizarse cuantas veces el analista desee hacerlo utilizando el botón DESORDENAR MAPA (Ver Figura 53), que ubica las fichas del rompecabezas en posiciones aleatorias, de forma que cada juego es un reto diferente.



Figura 53 Botón Desordenar Mapa de la Interfaz Gráfica del Perfil Analista

Si el analista no desea jugar puede utilizar el botón ARMAR ROMPECABEZAS (Ver Figura 54) que organiza automáticamente el rompecabezas.



Figura 54 Botón Armar Mapa de la Interfaz Gráfica del Perfil Analista

6.4.5.2 Elemento Lúdico para la Interfaz Gráfica del Perfil Director de Proyectos

El director de proyectos busca mantener información estratégica que haga crecer la productividad del grupo o grupos que dirige; por lo cual es necesario proveerlo de herramientas que le faciliten tomar sus apuntes.

La interfaz del director de proyectos cubre esta necesidad; proporcionando tres (3) cuadernillos de apuntes (Ver Figura 55) en los que el director podrá digitar los aspectos relevantes para analistas, diseñadores y programadores que se encuentran bajo su dirección.

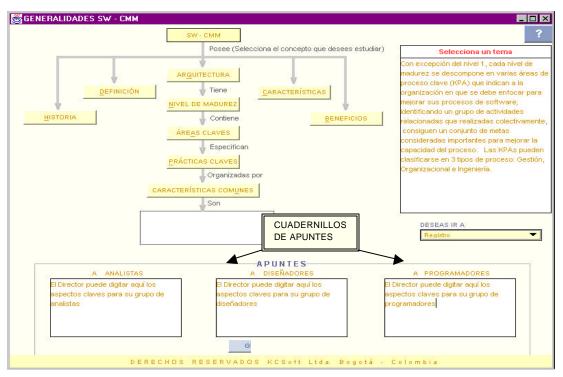


Figura 55 Elemento Lúdico (Cuadernillos de apuntes) de la Interfaz Grafica del Perfil Director de Proyectos

6.4.5.3 Elemento lúdico para la Interfaz Gráfica del Perfil Diseñador

Las actividades propias del diseñador lo presentan como un "Creador de Modelos", lo cual fue utilizado como base para el elemento lúdico que se le incluyó a su interfaz gráfica.

En ésta al diseñador se le proporciona un PANEL DE HERRAMIENTAS (Ver Figura 56), compuesto por:

- Botones: Son los elementos del mapa que al ser habilitados permiten acceder a alguna funcionalidad o información del tutor.
- Conectores: Son los enlaces que expresan el tipo de relación existente entre dos o más conceptos del mapa.
- Etiquetas: Son los conceptos que hacen parte del mapa conceptual, y ayudan al usuario a crear una estructura mental de la información, pero que al ser habilitados no proveen acceso a ninguna funcionalidad.
- Listas: Son los elementos que almacenan un conjunto de ítems que el usuario puede seleccionar para conocer sus explicaciones.

El diseñador debe arrastrar las herramientas y llevarlas a los componentes del mapa conceptual, ubicándolos en su posición correcta para así ir creando paulatinamente el mapa. A medida que el diseñador ubica correctamente las herramientas los elementos del mapa le son habilitados.

El mapa puede ser desarmado todas las veces que el diseñador lo desee utilizando el botón DESHACER MAPA. (Ver Figura 56)

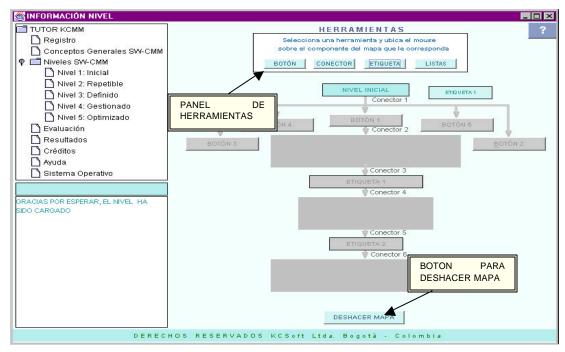


Figura 56 Elemento Lúdico (Herramientas) de la Interfaz Grafica Perfil Diseñador

6.4.5.4 Elemento Lúdico para la Interfaz Gráfica del Perfil Programador

Los programadores en su ambiente habitual interactúan con variables y atributos que modifican de acuerdo a las necesidades del software que estén codificando.

Teniendo en cuenta lo anterior la interfaz gráfica del programador tiene un PANEL DE PROPIEDADES (Ver Figura 57), en el cual se encuentran algunos atributos que pueden ser modificados al botón seleccionado. Estos atributos son:

Nombre: Es el identificador del botón seleccionado. Este no es editable.

Borde: Permite eliminar o colocar borde al botón seleccionado.

Alineación: Puede ser derecha, izquierda o centrado.

Texto: En la interfaz gráfica registro permite al usuario ingresar sus datos.

En el resto de interfaces gráficas permite adicionar un texto alternativo

que no es asignado al botón seleccionado.

