



ESCUELA SUPERIOR POLITECNICA DEL LITORAL

INSTITUTO DE CIENCIAS MATEMATICAS

INGENIERIA EN ESTADISTICA INFORMATICA

**“DISEÑO, DESARROLLO E IMPLEMENTACIÓN DE UNA HOJA
ELECTRÓNICA PARA EL CALCULO Y ANÁLISIS ACTUARIAL”**

TESIS DE GRADO

PREVIO A LA OBTENCION DEL TITULO DE:

INGENIERO EN ESTADISTICA INFORMÁTICA

PRESENTADO POR:

EDUARDO XAVIER FALCONES COLLANTES



GUAYAQUIL-ECUADOR

AÑO 2002

DEDICATORIA

A DIOS

A MIS PADRES

A MIS HERMANAS

A MI PADRINO

AGRADECIMIENTO

A todas las personas que de uno u otro modo colaboraron en la realización de este trabajo y especialmente en el Ing. Gaudencio Zurita Director de tesis, por su invaluable ayuda.

TRIBUNAL DE GRADUACIÓN

Ing. Felix Ramírez
Director del ICM

Ing. Gaudencio Zurita
DIRECTOR DE TESIS

Ing. Guillermo Gallego
VOCAL

Ing. Guillermo Vaquerizo
VOCAL

DECLARACION EXPRESA

“La responsabilidad del contenido de esta Tesis de Grado me corresponden exclusivamente, y el patrimonio intelectual de la misma a la ESCUELA SUPERIOR POLITECNICA DEL LITORAL.”

(Reglamento de Graduación de la ESPOL).

Miguel Flores Sánchez

INDICE GENERAL

	Pág.
RESUMEN	II
ÍNDICE GENERAL	III
ABREVIATURAS	IV
ÍNDICE DE FIGURAS	V
ÍNDICE DE TABLAS	VI
INTRODUCCIÓN	1

I. COMPOSICIÓN Y ELEMENTOS DE UNA HOJA ELECTRÓNICA

1.1. Elementos de una hoja electrónica	4
1.1.1. Hoja electrónica y celdas.....	4
1.1.2. Referencias y fórmulas.....	6
1.1.2.1. Un ejemplo con Microsoft Excel.....	7
1.1.3. Arrastrar celdas.....	9
1.1.4. Funciones predeterminadas.....	10
1.1.5. Aditamentos de una hoja electrónica.....	11
1.1.5.1. Tablas dinámicas.....	12
1.1.5.2. Macros de Excel.....	13
1.1.5.3. Filtros.....	13

II. UNA BREVE INTRODUCCIÓN A LA PROGRAMACIÓN ORIENTADA A

OBJETOS

2.1. La Abstracción.....	16
2.1.1. La evolución de la POO sobre la base del desarrollo del proceso de abstracción.....	17
2.1.1. Procedimientos.....	18
2.1.2. Módulos.....	19
2.1.3. Tipos abstractos de datos.....	20
2.1.4. Objetos.....	21
2.2. Programación estructurada.....	21
2.2.1. Desventajas.....	23
2.3. Orientación a objetos.....	23
2.3.1. Abstracción.....	24
2.3.2. Encapsulamiento.....	25
2.3.3. Modularidad.....	26
2.3.4. Jerarquía.....	26
2.3.5. Polimorfismo.....	27
2.4. Taxonomía de los lenguajes orientados a objetos.....	28
2.4.1. Lenguajes puros frente a híbridos.....	29
2.5. Beneficios de la tecnología de objetos.....	30
2.6. La programación orientada a objetos.....	31
2.6.1. Métodos y mensajes.....	33

2.6.2. Clases.....	35
2.6.3. Identificación de objetos.....	36
2.7. Comunicación entre objetos.....	38
2.7.1. Activación de objetos.....	39
2.7.2. Paso de mensajes.....	39
2.8. Estructura interna de un objeto.....	40
2.8.1. Atributos.....	40
2.8.2. Métodos.....	41
2.9. Herencias.....	42
2.9.1. Herencia Simple.....	44
2.9.2. Herencia Múltiple.....	44

III. MARCO TEÓRICO DE LAS MATEMÁTICAS ACTUARIALES

3.1. Definiciones.....	50
3.2. Distribuciones de supervivencia.....	50
3.2.1. Tiempo futuro de Supervivencia.....	54
3.2.2. Tiempo futuro de vida abreviada.....	56
3.2.3. Modelo de supervivencia quiebra.....	57
3.2.4. Tanto instantáneo f/q	60
3.2.5. Leyes de mortalidad teóricas.....	67
3.2.6. Esperanza de vida abreviada.....	71
3.2.7. Número total esperado de años de supervivencia.....	73

3.2.8. Tanto central de f/q a la edad.....	74
3.3. Modelos para las operaciones de seguros.....	75
3.3.1. Tasa de interés continua.....	76
3.3.2. Valor actuarial.....	79
3.3.3. Valores actuariales de modelos de seguros pagaderos al final año y al momento en que acaezca f/q	83
3.3.3.1. Modelos sin años de diferimiento.....	85
3.3.3.2. Modelos con años de diferimiento.....	91
3.3.3.3. Modelos con indemnización incremental y sin años de diferimiento.....	94
3.3.3.4. Modelos con indemnización incremental y con años de diferimiento.....	99
3.3.3.5. Otros modelos de seguros.....	101
3.3.4. Relación entre los modelos pagaderos al momento de f / q con aquellos pagaderos al final del año de f / q	105
3.4. Valores actuariales de las prestaciones para los casos de supervivencia.....	107
3.4.1. Valor Actuarial.....	108
3.4.2. Modelos de anualidades o rentas de vida.....	110
3.4.2.1. Modelos de rentas sin años de diferimiento.....	110
3.4.2.2. Modelos de rentas con años de diferimiento.....	115

3.4.2.3.	Otros modelos de rentas.....	118
3.4.3.	Capitalización actuarial.....	120
3.5.	Valores actuariales de las primas netas.....	121
3.5.1.	Valores actuarial para la prima neta, para operaciones de seguros y rentas vitalicias.....	124
3.5.2.	Valor actuarial, para operaciones de seguros y rentas vitalicias que incluye devolución o retención de primas....	128
3.5.3.	Primas netas fraccionadas.....	132
3.5.4.	Tipos de primas.....	134
3.6.	Reservas matemáticas.....	137
3.6.1.	Reserva matemática para distintas operaciones de seguros y rentas.....	140

VI. HERRAMIENTAS INFORMÁTICAS PARA EL DESARROLLO DE LA HOJA ELECTRÓNICA

4.1	Conceptos claves.....	143
4.1.1	Tiempo de diseño.....	143
4.1.2	Tiempo de ejecución.....	144
4.1.3	Interfaz de un solo documentos.....	144
4.1.4	Interfaz de múltiples documento.....	144
4.1.5	Entorno Integrado de Desarrollo (IDE).....	145
4.1.6	Interfaz de un objeto.....	146
4.1.7	Código reutilizable.....	147

4.1.8. Controles standart de visual basic.....	147
4.1.9. Referencias de visual basic.....	148
4.2 Ambito de las variables en VB.....	148
4.2.1 Declaración Dim.....	149
4.2.2 Declaración Public.....	149
4.2.3 Declaración Static.....	149
4.3 Módulos estándar y Módulos de clase.....	150
4.4 Creación de objetos personalizados en en Visual basic.....	150
4.4.1 Componenetes de un módulo de clase.....	151
4.4.1.1 Propiedades de una clase.....	151
4.4.1.2 Métodos de una clase.....	152
4.4.2 Colecciones de objetos.....	152
4.4.3 El modelo de objetos.....	154
4.4.4 Vida interna de un objeto.....	155
4.4.4.1 Variable de objeto.....	155
4.4.4.2 La palabra clave new.....	156
4.4.4.3 La palabra clave set.....	156
4.4.4.4 La palabra clave Nothing.....	156
4.4.4.5 La palabra clave Is.....	156
4.4.4.6 La palabra clave Typeof.....	157
4.4.4.7 La función TypeName.....	157
4.4.4.8 El evento Class_Iniatilate.....	157

4.4.4.9	El evento Class_Terminate.....	157
4.5	Acceso a datos.....	158
4.5.1	Conectividad ODBC.....	158
4.5.2	La arquitectura ActiveX data object (ADO).....	159
4.5.2.1	El objeto connection.....	160
4.5.2.2	El Objeto Recordset.....	161
4.5.2.3	El objeto Command.....	161
4.6	DLL Active X	161
4.6.1	El camino hacia Activa X.....	162
4.6.1.1.	Intercambio dinamico de datos.....	162
4.6.1.2.	Vinculación e Incrustación de Objetos 1.0.....	163
4.6.1.3.	Vinsulación e Incrustación de Objetos 2.0.....	164
4.6.1.4.	La tecnología Active X.....	164
4.6.1.5.	Identificadores de los objetos.....	165
4.6.2.	El modelo de objetos componente.....	166
4.6.2.1.	Caracterisiticas de los objetos COM.....	166
4.6.2.2.	Interfaces de automatización y envio.....	167
4.6.3.	Tipos de controloes y componenetes Active X.....	170
4.6.4.	Componentes de código Active X.....	171
4.6.5.	Ventajas de las DLL Active X y EXE Active X.....	174
4.6.6.	Desventajas de las DLL Active X y EXE Active X.....	175

V. DISEÑO DE LA HOJA ELECTRÓNICA ACTUARIAL

5.1. Análisis.....	178
5.1.1. Identificación de Inconvenientes y oportunidades.....	178
5.1.2. Definición del Sistema.....	182
5.1.3. Identificación de objetivos generales.....	184
5.1.3.1. Objetivos actuariales.....	184
5.1.3.2. Objetivos computacionales.....	190
5.1.4. Análisis de las necesidades del sistema.....	194
5.1.4.1. Diagrama de contexto.....	195
5.1.4.2. Diagrama cero.....	196
5.1.4.3. Módulo de análisis Uní modelo.....	199
5.1.4.3.1. Objetivos.....	199
5.1.4.3.2. Alcance.....	199
5.1.4.3.3. Diccionario de datos para Uni Modelos.....	200
5.1.4.3.3.1. Proceso: extraer tablas de mortalidad.....	200
5.1.4.3.3.2. Proceso: Generar modelos Actuariales.....	201
5.1.4.3.3.3. Proceso : Seleccionar Ley Teórica de Mortalidad..	202
5.1.4.3.3.4. Proceso: Calcular valores actuariales.....	202
5.1.4.3.3.5. Proceso: Analizar sensibilidad.....	203
5.1.4.3.3.6. Flujo: Requerimiento Tabla.....	203
5.1.4.3.3.7. Flujo: Características del modelo.....	204
5.1.4.3.3.8. Flujo: Objeto modelo actuarial.....	205

5.1.4.3.3.9. Flujo: Conjunto de tablas.....	205
5.1.4.4. Módulo de análisis Multi modelo.....	207
5.1.4.4.1. Objetivos.....	208
5.1.4.4.2. Alcance.....	209
5.1.4.4.3. Diccionario de datos.....	209
5.1.4.4.3.1. Proceso: Validar propiedades de los m modelos...	210
5.1.4.4.3.2. Proceso: Generar plan.....	211
5.1.4.4.3.3. Flujo: características de los m modelos.....	211
5.1.4.4.3.4. Flujo : m modelos actuariales generados.....	211
5.2. Diseño.....	212
5.2.1. Diseño de las DLL para modelos de seguros y rentas...	212
5.2.1.1. Diseño de Seguros.dll.....	215
5.2.1.1.1. Modelo de objeto.....	216
5.2.1.1.2. Las colecciones de objetos para modelos continuos y discretos.....	217
5.2.1.1.2.1. Método: CargarModelo.....	217
5.2.1.1.2.2. Método: BorrarModelo.....	218
5.2.1.1.2.3. Método: ObtenerModelo.....	218
5.2.1.1.2.4. Propiedad: CantDeModelos.....	218
5.2.1.1.3. Módulo de clase para modelo de seguro discreto....	219
5.2.1.1.3.1. Propiedad: PathBase.....	219
5.2.1.1.3.2. Propiedad: NombreTabla.....	220

5.2.1.1.3.3.	Propiedad: EdadInic.....	220
5.2.1.1.3.4.	Propiedad: Vigencia.....	220
5.2.1.1.3.5.	Propiedad: Diferido.....	220
5.2.1.1.3.6.	Propiedad: PeriodoDePago.....	221
5.2.1.1.3.7.	Propiedad: MomentoDePago.....	221
5.2.1.1.3.8.	Propiedad: TipoDeIndemnización.....	222
5.2.1.1.3.9.	Propiedad: Indemnización.....	222
5.2.1.1.3.10.	Propiedad: EdadMax.....	222
5.2.1.1.3.11.	Propiedad: TasaTécnica.....	223
5.2.1.1.3.12.	Propiedad: CdeltaAlfaGamma.....	223
5.2.1.1.3.13.	Método: ErrorParametros.....	223
5.2.1.1.3.14.	Método: MomentosSegurosDiscretos	224
5.2.1.1.4.	Módulo de clase para modelo de seguro continuo.....	225
5.2.1.1.4.1.	Propiedad: EdadInic.....	225
5.2.1.1.4.2.	Propiedad: Vigencia.....	226
5.2.1.1.4.3.	Propiedad: Diferido.....	226
5.2.1.1.4.4.	Propiedad: TipoDeLey.....	226
5.2.1.1.4.5.	Propiedad: CosntWABKN.....	226
5.2.1.1.4.6.	Propiedad: IndemnizaciónFuncional.....	227
5.2.1.1.4.7.	Propiedad: TasaTécnica.....	227
5.2.1.1.4.8.	Método: MomentosSeguroContinuo.....	227
5.2.1.1.4.8.1.	Bases de método de integración.....	228

5.2.1.1.4.8.2. Estructura del método.....	231
5.2.1.2. Diseño de Rentas.dll.....	232
5.2.1.2.1. Modelo de objeto.....	233
5.2.1.2.2. Las colecciones de objetos para modelos discretos..	234
5.2.1.2.2.1. Método: CargarModelo.....	234
5.2.1.2.2.2. Método: ObtenerModelo.....	235
5.2.1.2.2.3. Método: BorrarModelo.....	235
5.2.1.2.2.4. Propiedad: CantDeModelos.....	235
5.2.1.2.3. Módulo de clase para modelo de Rentas Discreto....	236
5.2.1.2.3.1. Propiedad: PathBase.....	236
5.2.1.2.3.2. Propiedad: NombreTabla.....	236
5.2.1.2.3.3. Propiedad: EdadInic.....	237
5.2.1.2.3.4. Propiedad: Vigencia.....	237
5.2.1.2.3.5. Propiedad: Diferido.....	237
5.2.1.2.3.6. Propiedad: PeriodoDePago.....	238
5.2.1.2.3.7. Propiedad: TipoDePago.....	238
5.2.1.2.3.8. Propiedad: Pago.....	238
5.2.1.2.3.9. Propiedad: EdadMax.....	239
5.2.1.2.3.10. Propiedad: TasaTécnica.....	239
5.2.1.2.3.11. Propiedad: CdeltaAlfaGamma.....	239
5.2.1.2.3.12. Propiedad: AbonoDeCuota.....	240
5.2.1.2.3.13. Método: ErrorParametros.....	240

5.2.1.2.3.14. Método: MomentosRentaDiscretos	240
5.2.2. Diseño de la interfaz visual para Maxim.....	242
5.2.2.1. Diseño del Modelo de Objetos.....	242
5.2.2.1.1. La colección grupo de hojas.....	243
5.2.2.1.1.1. Método: AñadirHoja.....	243
5.2.2.1.1.2. Método: RemoverHoja.....	244
5.2.2.1.1.3. Método: ObtenerHoja.....	244
5.2.2.1.1.4. Propiedad: Número de hojas.....	245
5.2.2.1.2. Módulo de clase para el objeto hoja electrónica.....	245
5.2.2.1.2.1. Propiedad: Nombre.....	245
5.2.2.1.2.2. Propiedad: Activa.....	246
5.2.2.1.2.3. Propiedad: Ventana.....	246
5.2.2.1.2.4. Propiedad: Clave.....	246
5.2.2.1.2.5. Propiedad: Negrita.....	246
5.2.2.1.2.6. Propiedad: Subrayado.....	247
5.2.2.1.2.7. Propiedad: Cursiva.....	247
5.2.2.1.2.8. Propiedad: Nuevo.....	247
5.2.2.1.2.9. Propiedad: HojaPath.....	247
5.2.2.1.2.10. Propiedad: NombredeHoja.....	248
5.2.2.1.2.11. Propiedad: RangoSeleccinado.....	248
5.2.2.1.2.12. Propiedad: GráficoAdjunto.....	248
5.2.2.1.2.13. Propiedad: Celda.....	248

5.2.2.1.2.14. Propiedad: CadenaEnBloque.....	249
5.2.2.1.2.15. Propiedad: FilaActiva.....	249
5.2.2.1.2.16. Propiedad: ColActiva.....	249
5.2.2.1.2.17. Método: Copiar.....	249
5.2.2.1.2.18. Método: Pegar.....	250
5.2.2.1.2.19. Método: Cortar.....	250
5.2.2.1.2.20. Método: Imprimir.....	250
5.2.2.1.2.21. Método: ImpConfMargeneSuperior.....	250
5.2.2.1.2.22. Método: ImpConfMargenIzquierdo.....	251
5.2.2.1.2.23. Método: ImpConfMargenInferior.....	251
5.2.2.1.2.24. Método: ImpConfMargenDerecho.....	251
5.2.2.1.2.25. Método: ImpEncabPiePagina.....	251
5.2.2.1.2.26. Método: OrientaciónDeHoja.....	252
5.2.2.1.2.27. Método: ImpEncabCol.....	252
5.2.2.1.2.28. Método: Imp EncabRow.....	252
5.2.2.1.2.29. Método: ImpLineasHojas.....	252
5.2.2.1.2.30. Método: Mostrar.....	253
5.2.2.1.2.31. Método: Guardar.....	253
5.2.2.1.2.32. Método: ColorBloque.....	253
5.2.2.1.2.33. Método: NegritaBloque.....	253
5.2.2.1.2.34. Método: CursivaBloque.....	254
5.2.2.1.2.35. Método: FuenteBloque.....	254

5.2.2.1.2.36. Método: TamañoFuenteBloque.....	254
5.2.2.1.2.37. Método: AlineaCentrado.....	254
5.2.2.1.2.38. Método: Alineaizquierda.....	255
5.2.2.1.2.39. Método: AlineaDerecha.....	255
5.2.2.1.2.40. Método: AñadirEliminarCol.....	255
5.2.2.1.2.41. Método: AñadirEliminarFila.....	255
5.2.2.1.3. La colección gráfico.....	256
5.2.2.1.3.1. Método: AñadirGrafico.....	256
5.2.2.1.3.2. Método: RemoverGrafico.....	256
5.2.2.1.3.3. Método: ObtenerGrafico.....	256
5.2.2.1.3.4. Propiedad: NúmerodeGraficos.....	256
5.2.2.1.4. Módulo de clase para el objeto gráfico.....	257
5.2.2.1.4.1. Propiedad: NombreGrafico.....	257
5.2.2.1.4.2. Propiedad: Ventana.....	257
5.2.2.1.4.3. Propiedad: NúmeroDeSeries.....	258
5.2.2.1.4.4. Propiedad: TipoDeGrafico.....	258
5.2.2.1.4.5. Propiedad: GraficoVisisble.....	258
5.2.2.1.4.6. Propiedad: NombreSeries.....	258
5.2.2.1.4.7. Propiedad: ConfigurarGrafico.....	259
5.2.2.1.4.8. Propiedad: NotaEjeX.....	259
5.2.2.1.4.9. Propiedad: NotaEjeY.....	259
5.2.2.2. Diseño de la Interfaz Visual.....	260

5.2.2.2.1.	Menú archivo.....	261
5.2.2.2.2.	Menú edición.....	262
5.2.2.2.3.	Menú análisis unimodelo.....	262
5.2.2.2.4.	Menú análisis multimodelos.....	263
5.2.2.2.5.	Menú ver.....	264
5.2.2.2.6.	Menú ventana.....	264
5.2.2.2.7.	Menú insertar.....	265
5.2.2.2.8.	Menú eliminar.....	265
5.2.2.2.9.	Menú ayuda.....	266
5.2.2.2.10.	El administrador de hojas y gráficos.....	266
5.2.2.2.11.	El formulario de la hoja electrónica.....	267
5.2.2.2.12.	El formulario del gráfico.....	268
5.2.2.2.13.	El formulario de configurar hoja.....	269
5.2.2.2.14.	El formulario para el asistente de gráficos.....	270
5.2.2.2.15.	Formulario para el módulo análisis uni modelo.....	271
5.2.2.2.15.1.	Formulario para modelos de seguros.....	271
5.2.2.2.15.2.	Formulario para modelos de rentas.....	272
5.2.2.2.15.3.	Formulario para sensibilizar modelos de seguros y rentas.....	273
5.2.2.2.16.	Formulario para el módulo análisis multi modelo....	274
5.2.2.2.16.1.	Formulario para iniciar un plan.....	275
5.2.2.2.16.2.	Formulario para generar plan de obligaciones.....	276

5.2.2.2.16.3. Formulario para generar el plan de pagos.....	276
5.2.2.216.4. Formulario para calcular primas y reservas.....	277
5.2.2.3. Otros módulos estándar y de clase.....	278
5.2.2.3.1. Módulo estándar para la manipulación de hojas.....	278
5.2.2.3.1.1. Procedimiento: InsertarNuevaHoja.....	278
5.2.2.3.1.2. Procedimiento: InsertarHojaAbierta.....	278
5.2.2.3.1.3. Procedimiento: EliminarHoja.....	279
5.2.2.3.1.4. Procedimiento: PreguntarPorGuardar.....	279
5.2.2.3.1.5. Procedimiento: InsertarNombreNuevaHoja.....	279
5.2.2.3.2. Módulo estancar para la manipulación de gráficos.....	280
5.2.2.3.2.1. Procedimiento: AñadirGraficoArbolCole.....	280
5.2.2.3.2.2. Procedimiento: BorrarGrafico.....	280
5.2.2.3.3. Módulo de clase para primas y reservas.....	280
5.2.2.3.3.1. Propiedad: PlandDeObligacionesSeg.....	281
5.2.2.3.3.2. Propiedad: PlanDeObligacionesRent.....	281
5.2.2.3.3.3. Propiedad: PlanDePagos.....	281
5.2.2.3.3.4. Propiedad: TasaPlan.....	281
5.2.2.3.3.5. Propiedad: TablaPlan.....	281
5.2.2.3.3.6. Método: CacularPrimaNeta.....	282
5.2.2.3.3.7. Método: CalcularPrimaComercial.....	282
5.2.2.3.3.8. Método: CalcularPrimaInventario.....	282

6.5.3.2. Sensibilizar modelos de seguros a través de la edad del asegurado.....	311.
6.5.3.3. Sensibilizar modelos a través del diferimiento.....	316
6.5.3.4. Sensibilizar modelos a través de la vigencia.....	320
6.5.4. Generación de modelos discretos de rentas.....	322
6.5.5. Análisis de sensibilidad de modelos de rentas.....	324
6.5.5.1. Modelos de renta sensibilizados a través de la tasa de interés.....	326
6.5.5.2. Modelos de rentas sensibilizados a través de la edad del asegurado.....	331
6.5.5.3. Modelos de rentas sensibilizados a través de los años de diferimiento.....	333
6.5.5.4. Modelos actuariales sensibilizados a través de la vigencia del modelo.....	335
6.6. Módulo para el análisis Multi – modelos.....	337
6.6.1. Inicio de un plan.....	337
6.6.2. Generar plan de obligaciones.....	339
6.6.3. Generar plan de beneficios.....	341
6.6.4. Interfaz para el calculo de primas y reservas de un plan..	342
6.6.4.1. Prima Neta y reserva matemática de la prima.....	344
6.6.4.2. Prima de inventario y reserva matemática de la prima...	348
6.6.4.3. Prima comercial y reserva matemática para la prima.....	352

CONCLUSIONES Y RECOMENDACIONES

BIBLIOGRAFIA

ABREVIATURA

MSD	Modelo de seguro discreto
MSC	Modelo de seguros continuo
MRD	Modelo de renta discreta
MRC	Modelo de renta continuo
F/q	Evento de fallecimiento o quiebra
(x)	Edad de x años de edad
u.m	Unidad monetaria

SIMBOLOGIA

$S(x)$	Función de supervivencia
$F(x)$	Función de perdida
T_x	Tiempo futuro de supervivencia
${}_t q_x$	Función de distribución del tiempo futuro de supervivencia
${}_k / q_x$	Tiempo futuro de vida abreviada
l_x	Número de personas que se espera sobreviva a la edad x
m_x	Tase de mortalidad
e_x	Esperanza de vida abreviada
e_x^o	Esperanza de vida completa
v	Factor de actualización
m_x	Tasa central de f/q

SUMARIO

Capitulo 1

Composición y elementos de un Hoja Electrónica

Capitulo 2

Una breve introducción a la programación Orientada a Objetos

Capitulo 3

Marco Teórico de las Matemáticas Actuariales

Capitulo 4

Herramientas informáticas para el desarrollo de la hoja electrónica

Capitulo 5

Diseño de la hoja electrónica y componente Active X

Capitulo 6

La hoja electrónica actuarial en tiempo de ejecución

INDICE DE FIGURAS

		Pág
Figura. 1.1.	Hoja de Microsoft Excel 7.0.....	5
Figura 1.2.	Media y Proporciones por Estratos Económicos.....	7
Figura 1. 3.	Resultados de la Formula.....	8
Figura 1.4.	Arrastrar celdas.....	9
Figura 1.5.	Funciones de Microsoft Excel.....	11
Figura 1.6.	Tablas dinámicas de Excel.....	12
Figura 2.1.	Partes de un módulo.....	19
Figura 2.2.	Taxonomía de los lenguajes OO.....	29
Figura 2.3.	Mensajes entre objetos.....	34
Figura 2.4.	Atributos de un objeto.....	37
Figura 2.5.	Mensajes internos.....	42
Figura 2.6.	Herencia.....	43
Figura 2.7	Herencia Simple.....	44
Figura 2.8.	Herencia Múltiple.....	45
Figura 3.1.	Estructura de un modelo actuarial.....	48
Figura 3.2.	Rango de fallecimiento o quiebra de un ente, entre 0 y X años.....	51
Figura 3.3.	Rango de sobre vivencia de un ente entre 0 y X años de edad.....	52
Figura 3.4.	Intervalo de fallecimiento o quiebra de un ente	53
Figura 3.5.	Probabilidad que un individuo (x) sobreviva t años más y fallezca después de t+n años.....	56
Figura 3.6.	Diferencia en el pago entre seguros continuos y discretos con indemnización incremental.....	95
Figura 4.1.	Entorno integrado de desarrollo de Visual Basic.....	145
Figura 4.2.	Controles standard.....	147
Figura 4.3.	Estructura de un objeto de colección.....	153
Figura 4.4.	Modelo de objetos para una universidad.....	154
Figura 4.5.	Modelo de objeto ADO.....	159
Figura 4.6.	Gráfico de Excel incrustado en un documento de Word.....	163
Figura 4.7.	Estructura de un módulo de clase.....	169
Figura 4.8	Organización de la memoria con dos aplicaciones ejecutándose.....	172
Figura 4.9.	Esquema del llamado de una función contenida en un componente EXE Active X.....	173
Figura 5.1.	Valor Actuarial para diferentes tasas de interés.....	180
Figura 5.2.	El ciclo de la información.....	183
Figura 5.3.	Asignación de una tabla de mortalidad a un modelo.....	185
Figura 5.4.	Asignación de un modelo a una ley de mortalidad.....	186
Figura 5.5.	Modelos continuos con indemnización Funcional.....	188
Figura 5.6.	Esquema de un Plan (Plan A).....	189
Figura 5.7.	Hoja electrónica de Microsoft Excel.....	190
Figura 5.8.	Asistente gráfico de Excel.....	191
Figura 5.9.	Esquema de código reutilizable.....	193
Figura 5.10.	Visión General del Sistema.....	195

ÍNDICE DE TABLAS

		Pág
Tabla I	Simbología actuarial para planes.....	126
Tabla II	Valor presente de las obligaciones futuras.....	138
Tabla III	Tablas de mortalidad asignadas a los modelos actuariales.....	213
Tabla IV	Las tablas de mortalidad asignadas a los modelos.....	296
Tabla V	Modelos discretos considerados.....	297
Tabla VI	Modelos continuos considerados.....	298
Tabla VII	Valores actuariales para los modelos de seguros discretos.....	299
Tabla VIII	Tabla de valores actuariales para el modelo MSC –1.....	301
Tabla IX	Valores Actuarial para el modelo MSC –2.....	302
Tabla X	Valor Actuarial para un modelo asignado a una ley de Moivre, para varias indemnizaciones funcionales.....	303
Tabla XI	Indemnización del modelo MSC-3 continuo asignado a una indemnización funcional cuadrática.....	304
Tabla XII	Valores Actuariales de los modelos MSD-1, MSD-2, MSD-5 considerando distintas tasas de interés.....	308
Tabla XIII	Valores Actuariales para los modelos de seguros, variando la edad del asegurado.....	311
Tabla XIV	Modelos actuariales adicionales de dote puro y de indemnización incremental.....	313
Tabla XV	Valor actuarial para los modelos de dote puro y de indemnización incremental.....	314
Tabla XVI	Valores actuariales y riesgos para los modelos de seguro, variando los años de diferimiento.....	316
Tabla XVII	Valores actuariales para los modelos de seguros variando la edad de vigencia.....	320
Tabla XVII	Valores actuariales para los modelos de seguros, variando la edad de vigencia.....	320
Tabla XVIII	Modelos de rentas discretos considerados.....	323
Tabla XIX	Valores actuariales para modelos de rentas discretos, variando la tasa de interés.....	326
Tabla XX	Valor actuarial para los modelos MRD-3, MRD-4, variando la tasa de interés.....	329
Tabla XXI	Valores actuariales para modelos de rentas discretos, variando la edad del asegurado.....	331
Tabla XXII	Valor actuarial para los modelos de rentas, variando los años de diferimiento.....	333
Tabla XXIII	Valor actuarial para los modelos de rentas discretos, variando la vigencia de los modelos.....	335
Tabla XXIV	Características iniciales para plan a generar.....	338
Tabla XXV	Modelos para plan de obligaciones.....	340
Tabla XXVI	Modelos asignados al plan de pagos.....	342
Tabla XXVII	Prima neta para distintas tasas de interés.....	344

Tabla XXVIII	Reservas matemáticas para la prima neta, evaluada en distintas tasas de interés.....	346
Tabla XXIX	Primas de inventario calculadas a distintas tasas de interés.....	348
Tabla XXX	Reserva matemática para la prima de inventario, para diferentes tasas de interés.....	350
Tabla XXXI	Primas comerciales calculadas a diferentes tasas de interés.....	352
Tabla XXXII	Reserva matemática para la prima comercial, para diferentes tasas de y edades.....	354

CAPITULO I

1. COMPOSICIÓN Y ELEMENTOS DE UNA HOJA ELECTRÓNICA

La eficiencia es un elemento de gran importancia en el diseño, elaboración, y análisis de un proyecto, este tipo de eficiencia brinda las hojas electrónicas, especialmente para los diferentes tipos de cálculos que se han de realizar.

Las hojas electrónicas se han ido desarrollando a la par con la tecnología informática, las primeras estaban orientadas principalmente al análisis contable, aquí tenemos QPRO que era una hoja electrónica cuya plataforma era el sistema operativo DOS por ende la interfaz no

era muy amigable, y se tenía que combinar algunas teclas para realizar algún tipo de evento.

Las hojas de cálculo actuales como Excel de Microsoft presentan diferencias significativas con las anteriores, aunque su diseño sigue orientado al análisis y cálculo el contable – financiero, estas ya presentan ciertos tipos de aditamentos que permiten que profesionales de áreas distintas a las contables las empleen.

Estas hojas electrónicas presentan cierto grado de flexibilidad, puesto que un usuario puede personalizar sus requerimientos utilizando código de programación (macros), permitiendo desglosar un problema aprovechando la interfaz que presentan estas hojas.

Como se mencionó antes la flexibilidad de las hojas actuales no es completa, aunque en ciertas circunstancias la interfaz es la adecuada, no podemos llegar eficientemente a los resultados de algún objetivo planteado, como es el caso del cálculo y análisis del extenso campo de las matemáticas actuariales.

1.1. Composición y elementos de una hoja electrónica

Aunque las Hojas Electrónicas de la actualidad como Microsoft Excel están más orientadas al aspecto contable y en menor grado al análisis estadístico financiero, la mayor parte de las empresas aseguradoras que existen en Guayaquil utilizan esta herramienta informática para analizar Reservas, primas, tablas de mortalidad, ya que presentan la interfaz adecuada y la posibilidad de desarrollar Macros que se ajusten a los requerimientos del Investigador.

En tiempo de ejecución la hoja electrónica diseñada en este proyecto presenta características definidas de una Hoja electrónica, por esto vamos a mencionar los elementos principales que componen una.

1.1.1. Hoja electrónica y celdas

Una hoja electrónica es un arreglo bidimensional compuesta de un gran número de filas y columnas (en el caso de Excel menos de 6600 filas y 300 columnas), cuyas intersecciones permiten el

ingreso de información en diversos tipos de formatos, a éstas intersecciones se las denomina Celdas.

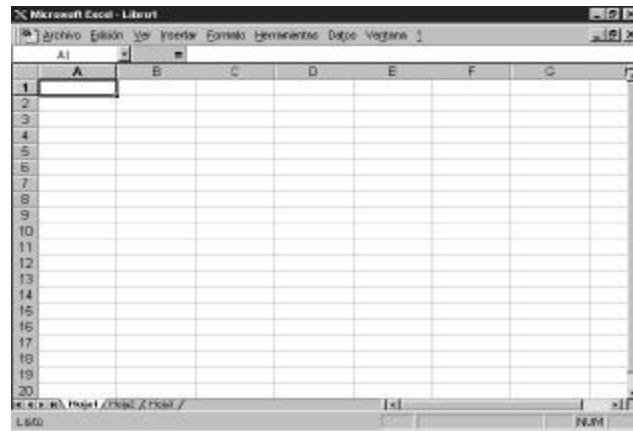


Figura. 1.1. Hoja de Microsoft Excel 7.0

Los tipos de formatos que puede presentar una celda son:

- Número
- Fecha
- Hora
- Fracción
- Científica
- Porcentajes
- Texto
- Fórmulas

Una celda al ser portadora del resultado de una fórmula, puede necesitar como parámetros de entrada la ubicación de otras celdas.

1.1.2. Referencias y formulas

Al hablar de referencia se refiere a la necesidad de información que tiene una celda “y” de una celda “x” o de varias celdas “ X_1, X_2, \dots, X_n ”, entonces tenemos que el valor de la celda “y” depende de los valores que tengan las celdas a las que se están haciendo referencia.

Una fórmula es una función definida por el usuario, donde los valores de sus parámetros las obtiene de una o varias celdas, aquí existe una iteración entre las celdas independientes y dependiente.

$$Y = g(X_1, X_2, \dots, X_n) \quad (1.1)$$

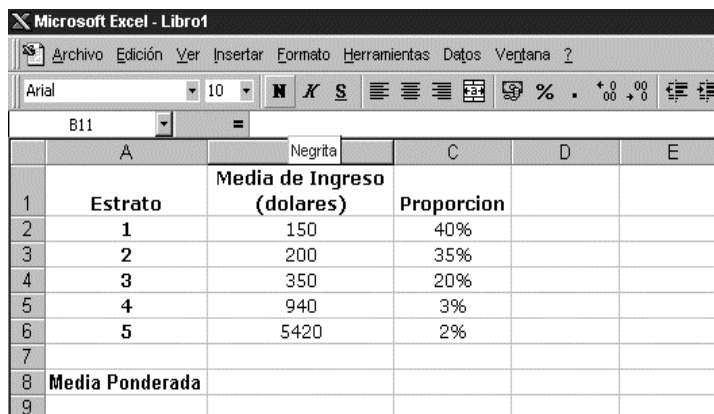
En ocasiones no es necesario que una fórmula haga referencia a celdas específicas de una Hoja Electrónica, ya que se puede definir una fórmula de la siguiente forma.

$$Y = (10+5)*4-2 \quad (1.2)$$

Vemos que “y” no interactúa con otras celdas de la hoja electrónica, la celda “y” toma un valor fijo y sólo es posible modificarla cambiando la fórmula dentro de la celda.

1.1.2.1. Un ejemplo con microsoft excel

En la figura dos se observa que existen dos columnas con información, en la columna A tenemos la media de ingreso por estrato social de una ciudad y en la columna B la proporción que representa cada estrato con respecto a toda la ciudad.



The screenshot shows a Microsoft Excel window titled 'Microsoft Excel - Libro1'. The menu bar includes Archivo, Edición, Ver, Insertar, Formato, Herramientas, Datos, Ventana, and ?. The toolbar shows the font set to Arial, size 10, with options for Bold (N), Italic (K), and Underline (S). The active cell is B11, containing an equals sign (=). The spreadsheet data is as follows:

	A	B	C	D	E
		Media de Ingreso (dolares)	Proporción		
1	Estrato				
2	1	150	40%		
3	2	200	35%		
4	3	350	20%		
5	4	940	3%		
6	5	5420	2%		
7					
8	Media Ponderada				
9					

Figura 1.2 Media y Proporciones por Estratos Económicos

Lo que deseamos obtener es la media ponderada del Ingreso para esta ciudad, entonces la sintaxis de la fórmula que se debe de ingresar en la Celda "B8" es:

$$=B2*C2+B3*C3+B4*C4+B5*C5+B6*C6 \quad (1.3)$$

Como se muestra en la siguiente figura

	A	B	C	D	E
1	Estrato	Media de Ingreso (dolares)	Proporcion		
2	1	150	40%		
3	2	200	35%		
4	3	350	20%		
5	4	940	3%		
6	5	5420	2%		
7					
8	Media Ponderada	336,6			
9					

Figura 1. 3 Resultados de la Formula

1.1.3. Arrastrar celda

Una característica de una hoja electrónica es la simplicidad que brinda en los cálculos que tiene que realizar el usuario para obtener un resultado. Ahora el usuario está interesado en obtener distintos valores pero utilizando la misma fórmula, utilizando el ejemplo anterior estaría interesado en obtener la media ponderada de todas las ciudades principales de un país (supongamos que el país tiene 3 ciudades principales) que se encuentra estratificada por nivel económico como se muestra en la siguiente figura.

	A	B	C	D	E	F	G	H
1			Proporciones			Medias de Ingresos		
2	Estrato	Ciudad1	Ciudad2	Ciudad3	Ciudad1	Ciudad2	Ciudad3	
3	1	50%	45%	60%	111	115	98	
4	2	25%	25%	10%	215	200	150	
5	3	10%	25%	20%	450	354	448	
6	4	10%	3%	9%	845	951	785	
7	5	5%	2%	1%	1050	1500	980	
8								
9			Ciudad1	Ciudad2	Ciudad3			
10	Media Ponderada		291,25	248,78	243,85			
11								

Figura 1.4 Arrastrar celdas

Como se observa no es necesario volver a colocar la fórmula de media ponderada para las ciudades restantes, basta con arrastrar

una celda donde ya este definida la fórmula para obtener la media ponderada correspondiente a cada ciudad.

1.1.4. Funciones predeterminadas

Una hoja electrónica posee ya funciones para tareas específicas como la función MÁXIMO(Rango) que obtiene el máximo entre un rango de valores, MINIMO(Rango) obtiene un valor mínimo entre un rango de valores, SUMA(Rango) obtiene la suma de un rango de valores, TIR(Rango) obtiene la tasa interna de retorno de un flujo de caja, etc. Estas funciones también pueden ser parte de una fórmula o pueden estar contenidas en otras funciones por ejemplo:

$$\begin{aligned} &= B1 * \text{MÁXIMO}(C1:C100) \\ &= \text{SUMA}(\text{MÁXIMO}(C1:C20), \text{MÁXIMO}(D1:D20), \\ &\quad \text{MINIMO}(E1;E20)) \end{aligned} \quad (1.4)$$

En la siguiente figura se presenta algunas de las funciones que posee Microsoft Excel.

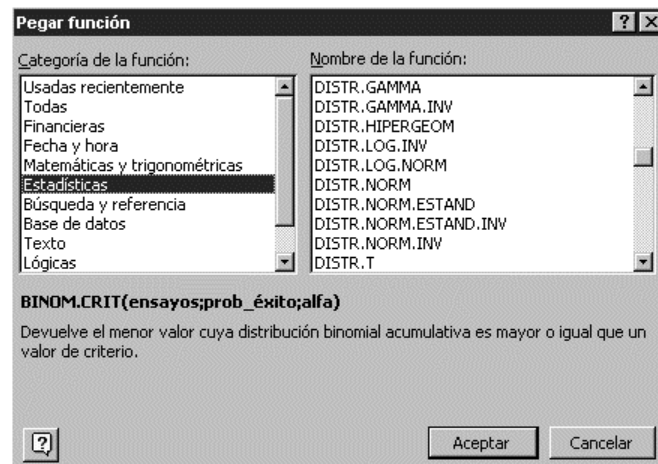


Figura 1.5 Funciones de Microsoft Excel

1.1.5. Aditamentos de una hoja electrónica

En forma general todo lo mencionado anteriormente es lo que caracteriza a una hoja electrónica, pero existen aditamentos especiales que se adhieren a estas hojas para brindar una mayor facilidad en la manipulación de los datos, Microsoft Excel que es nuestra hoja electrónica de referencia posee varios aditamentos que hacen más poderosa esta herramienta informática, entre los más destacables se tienen:

- Tablas Dinámicas
- Macros de Excel
- Filtros

1.1.5.1. Tablas dinámicas

Las tablas dinámicas permiten obtener ciertas características o eventos de una o más variables que están ingresados en la hoja de cálculo por columnas según los requerimientos del usuario por ejemplo supongamos que hemos realizado una encuesta a 20 personas en una universidad y hemos obtenido dos características de estas personas el sexo y la edad, se quiere saber cuantas mujeres y cuantos hombres son mayores de edad. En la siguiente figura se muestra el asistente de tablas dinámicas para Excel.



Figura 1.6 Tablas dinámicas de Excel

1.1.5.2. Macros de excel

Cuando el usuario desea personalizar un proceso, este lo puede programar en un lenguaje basado en objetos como lo es Visual Basic creando macros que faciliten la solución de su problema.

1.1.5.3. Filtros

Como su nombre lo indica un filtro nos permite filtrar los datos para poder trabajar solo con la información requerida.

CAPITULO II

2. BREVE INTRODUCCIÓN A LA ORIENTACIÓN A OBJETOS

En el ambiente donde desarrollamos nuestras actividades diarias podemos identificar dos clases de objetos diferentes, tangibles e intangibles, algunos de estos con propiedades y actividades bien definidas.

En muchos de los problemas que el hombre desea solucionar están presentes objetos de estas dos clases, por ejemplo supongamos que

un administrador de red desea mejorar la saturación que tiene su servidor cuando las estaciones de trabajo envían sus documentos en forma simultánea a la única impresora conectada a la red, para esto el administrador tiene que analizar todos los posibles factores que producen esta saturación, como pueden ser la velocidad de la red, el tipo de cableado, la cola de paquetes, etc. Aquí podemos identificar la velocidad de la red, y la cola de paquetes como objetos intangibles, como una propiedad del paquete de datos se tiene la longitud y como una propiedad de la cola el valor esperado del tiempo de espera del paquete en la cola, el cableado es un objeto tangible y unas de sus propiedades puede ser el color, la capacidad, etc.

La programación orientada a objetos (POO) es el resultado del proceso de separar las cualidades de un objeto, para considerarlo en su pura esencia, esto es el desarrollo del proceso de la abstracción. Esto ayuda sin duda a simplificar sistemas complejos, proporcionando conceptos y herramientas con las cuales se modela el mundo real tan fielmente como sea posible.

2.1. Abstracción

La abstracción es el proceso por el cual se toman las partes esenciales de un objeto para simplificar su significado y obtener así la esencia pura de este.

Entonces se puede mencionar que un algoritmo es una descripción abstracta de un procedimiento real, los algoritmos trasladados a un lenguaje computacional nos ayudan a simular un fenómeno físico o una actividad humana.

La abstracción y el lenguaje de programación tienen una relación biunívoca, puesto que nosotros realizamos una abstracción de un problema real y traducimos las ideas a un lenguaje de programación y el lenguaje realiza una abstracción del comportamiento del computador.

El desarrollo del nivel de la abstracción ha ido de la mano con el desarrollo de los lenguajes de programación, así en los años 50, el mecanismo de abstracción era el lenguaje de máquina o lenguaje ensamblador, los lenguajes de alto nivel de la actualidad nos ofrecen un nuevo nivel de abstracción.

2.1.1. La Evolución de la POO sobre la base del desarrollo del proceso de abstracción

El proceso de abstracción fue evolucionando desde la aparición de los primeros lenguajes de programación, el nivel más bajo de abstracción en la programación fue en los inicios del lenguaje ensamblador, este consistía en abstracciones diseñadas para evitar que los programadores tuvieran que recordar las secuencias de bits que componen las instrucciones de un programa, tras este lenguaje surgieron los lenguajes de alto nivel, esto supuso un nuevo nivel de abstracción, este tipo de lenguajes permitió que los programadores superen los tecnicismos de la arquitectura de un computador, como consecuencia no había la necesidad de desarrollar un programa para determinado tipo de máquina y hacer ciertas modificaciones para otros tipos de computadores, al agrupar secuencias de sentencias en los lenguajes de alto nivel formamos los procedimientos. Surgió así la programación estructurada el cual alienta el uso de abstracciones de control.

El método más idóneo para controlar la complejidad fue aumentar los niveles de abstracción, así tenemos que las técnicas

orientada a objetos fueron el resultado de una larga evolución histórica que incluyen las estructuras de control, procedimientos, módulos, tipos abstractos de datos, hasta llegar a los objetos.

2.1.2. Procedimientos

Los primeros mecanismos de abstracción utilizados en los lenguajes de programación fueron los procedimientos, los cuales permitían reutilizar el código una y otra vez sin necesidad de volver a escribirlo, esto suponía una mejora significativa en la eficiencia de los programas.

Aunque con este mecanismo se consideró por primera vez la posibilidad de ocultación de la información, no fue efectivo para este propósito debido a sus limitaciones, por ejemplo supongamos que declaramos una cola con las siguientes rutinas Introducir(), Sacar(), etc. Es obvio que los datos de la cola deben ser globales y no locales ya que las modificaciones que se realizan en una rutina deben de ser consideradas por las demás, al ser las variables globales no existe un método para limitar la accesibilidad o visibilidad de dichas variables.

2.1.3. Módulos

Este mecanismo en parte resuelve las limitaciones que tenían los procedimientos inicialmente, puesto que esta técnica proporciona la posibilidad de dividir sus datos y procedimientos en una parte privada que solo es accesible dentro del módulo y una parte pública, que es accesible fuera del módulo, pudiéndose definir cualquier tipo de datos en cualquiera de las dos divisiones como se ilustra en la siguiente figura.

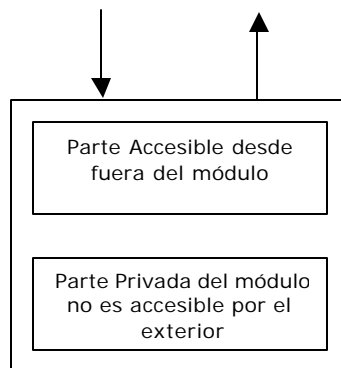


Figura 2.1. Partes de un módulo

Una de las limitaciones de los módulos es que no permiten hacer instanciación, que es la capacidad de hacer múltiples copias de la zona de datos.

2.1.4. Tipos abstractos de datos

Un tipo abstracto de datos o TAD, es un tipo personalizado por el programador que se pueden manipular de una manera similar a los tipos de datos definidos por el sistema, al igual que los tipos de datos del sistema este se mueve por un conjunto de valores legales de datos y un número de operaciones primitivas, por ejemplo la pila es un TAD cuyas operaciones definidas son Meter, Sacar, Pila vacía, Pila llena, Nueva pila, etc .

Para implementar un TAD frecuentemente se utilizan los módulos, para construir un tipo abstracto de datos se debe poder:

1. Exponer una definición del tipo
2. Hacer disponibles un conjunto de operaciones que se puedan utilizar para manipular instancias de ese tipo.
3. Proteger los datos asociados con el tipo.
4. Permitir instancias múltiples del tipo

2.1.5 Objetos

Un objeto es algo al cual se le pueden hacer cosas. Este tiene un estado, comportamiento e identidad; la estructura y comportamiento de objetos similares se definen en su clase común.

Los mecanismos básicos de orientación de objetos son: Objetos, mensajes y métodos, clases instancia y herencia. Una de las características fundamentales es el paso de mensajes entre objetos, de aquí se derivan las ideas de herencia y polimorfismo.

La herencia permite la utilización de un mismo código por diferentes tipos de datos, esto reduce considerablemente el tamaño del código.

El polimorfismo permite enviar un mismo mensaje a diferentes tipos de objetos y que estos tengan a su vez un comportamiento diferente entre ellos.

2.2. Programación Estructurada

Esta técnica de programación tiene sus principios en la década de los setenta y en la actualidad es una de las más utilizadas.

Esta técnica comienza descomponiendo el programa en módulos más pequeños y manejables llamadas funciones, éstas permiten tener un mayor control de la totalidad del programa, así como la detección de errores en forma más veloz, pero cuando se detectan estos errores pueden causar una pérdida de tiempo valioso ya que es posible que existan variables que sean utilizadas por varias funciones en este caso habría que modificar las sentencias en todas estas funciones.

En un programa estructurado es suficiente conocer que un procedimiento dado realiza una tarea específica, como se realiza esta tarea no es importante si no, que recibe de argumento y cuales son los resultados.

En un programa estructurado los datos locales se ocultan dentro de las funciones, mientras que los datos globales se los pasa como argumentos a las funciones, esto en algunos casos es una

desventaja ya que la disposición de los datos no se pueden cambiar sin modificar todas las funciones que acceden a ellos.

2.2.1. Desventajas

Como ya se mencionó la programación estructurada tiene sus desventajas entre éstas se pueden mencionar:

- Los programas basados en funciones son muy difíciles de diseñar, puesto que sus componentes principales no modelan bien el mundo real.
- Los errores de comunicación entre miembros de un equipo de programadores pueden conducir a gastar tiempo en reescritura.
- Creación de nuevos tipos de datos

2.3. Orientación a Objetos

La programación orientada a objetos permite una representación más directa del mundo real en el código, la orientación a objetos trata de cumplir los requerimientos del usuario, esto se lo realiza

mediante la modelización del mundo real y el soporte fundamental es el modelo de objetos cuyos elementos de mayor relevancia son: La abstracción, encapsulamiento, modularidad, jerarquía y polimorfismo. También existen en este modelo 3 elementos secundarios como son: tipificación, concurrencia, persistencia, al decir secundario no se quiere mencionar que estos elementos no son útiles, pero no son esenciales.

2.3.1. Abstracción

Una abstracción denota las características esenciales de un objeto que lo distinguen de todos los demás tipos de objetos y proporciona así fronteras conceptuales nítidamente definidas respecto a la perspectivas del observador, por ejemplo suponga que dos personas se dirigen a comprar un vehículo (un señor y una señora), por casualidad observan el mismo vehículo y de él obtienen sus características esenciales según sus perspectivas, entonces el señor le dará más importancia a la potencia del vehículo, la velocidad máxima que alcanza el vehículo en 20 segundos, etc. Mientras que la señora se fijara más en lo espacioso del carro, en el color, en las formas etc.

El elemento clave de la programación orientada a objetos es la clase, esta se puede definir como una descripción abstracta de un grupo de objetos los cuales se diferencia por su estado y operación.

2.3.2. Encapsulamiento

El encapsulamiento es el proceso de almacenar en un mismo compartimiento los elementos de una abstracción que constituye su estructura y su comportamiento, ésta propiedad permite asegurar que el contenido de la información de un objeto esta oculto al mundo exterior.

La encapsulación permite dividir los programas en módulos, estos se implementan mediante clases, de forma que la clase representa la encapsulación de una abstracción. Esto significa que cada clase debe de poseer dos partes: una interfaz y una implementación. La interfaz solo captura la vista externa y la implementación contiene la representación de la abstracción, así como los procedimientos que realizan el comportamiento deseado.

2.3.3. Modularidad

Consiste en dividir un programa en partes o módulos más pequeños independientes unos de los otros pero que tienen conexión con otros módulos, esto es que cada módulo se lo pueda compilar separadamente sin problemas y al juntarlos todos en un programa los módulos puedan interactuar entre sí.

2.3.4. Jerarquía

La jerarquía es una clasificación u ordenación de abstracciones. Frecuentemente un conjunto de abstracciones forma una jerarquía, y la identificación de esas jerarquías en el diseño simplifica en gran medida la comprensión de problema.

Las dos jerarquías más importantes en un sistema complejo son su estructura de clases (La jerarquía de clases) y su estructura de objetos (La jerarquía de partes).

2.3.5. Polimorfismo

Literalmente hablando polimorfismo es la propiedad que tienen algunos cuerpos de cambiar de forma conservando su naturaleza. Acoplado esta definición a la POO, tenemos que el polimorfismo permite referirse a objetos de clases diferentes mediante el mismo elemento de programa y realizar la misma operación de formas diferentes.

Consideremos la clase transportación, de aquí podemos obtener subclases, las cuales pueden ser transportación terrestre, transportación aérea, transportación fluvial, a dos objetos diferentes se les manda un mensaje "llevar a los pasajeros desde su origen hasta su destino", cada objeto lo entenderá de la misma manera pero para lograr su objetivo se valeran de diferentes medios, si es el transporte aéreo se valdrá del aire para lograrlo, pero si el objeto pertenece al transporte terrestre utilizará el asfalto de las carreteras.

El polimorfismo implica la posibilidad de tomar un objeto de un tipo, e indicarle que ejecute una operación, esta se lo realizará de diferentes formas, según sea el objeto de donde se aplica.

2.4. Taxonomía de los lenguajes Orientados a Objetos

La taxonomía es la ciencia que trata sobre los principios de la clasificación, una taxonomía de lenguajes de programación con propiedades de orientación a objetos fue creada por Wegner. La clasificación incluye los siguientes grupos:

Basado en Objetos: Un lenguaje de programación es basado en objetos si su sintaxis y semántica soporta la creación de objetos que tiene las propiedades antes ya mencionadas.

Basado en Clases: Si un lenguaje de programación es basado en objetos y soporta además la creación de clases, se considera basado en clases.

Orientado a Objetos: Un lenguaje de programación orientados a objetos es un lenguaje basado en clases que soporta también herencia.

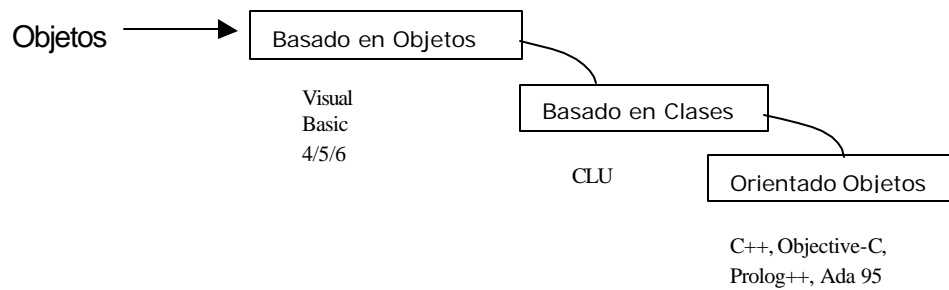


Figura 2.2. Taxonomía de los lenguajes OO

2.4.1. Lenguaje de programación orientados a objetos puros frente a híbridos

Un Lenguaje puro es aquel que fue diseñado únicamente para soportar el paradigma de la programación orientada a objetos, consta de métodos, objetos, clases entre los más populares tenemos Eiffel, Smalltalk. Por el contrario un Lenguaje de Programación Híbrido soporta otros paradigmas de programación tales como el estructurado, funcional, etc.

Los lenguajes híbridos se construyen a partir de otros lenguajes existentes, tales como C o Pascal.

Cada tipo de lenguajes tiene sus ventajas y desventajas. Los lenguajes puros pueden ser más potentes y flexibles, ya que utilizan todas las ventajas de la tecnología orientada a objetos. Pero el precio que tienen que pagar los lenguajes puros debido a su potencia y flexibilidad es que no son tan rápidos como los lenguajes híbridos y son difíciles de codificar en toda clases de operaciones fundamentales. Esto les hace perder eficiencia en tiempo de ejecución.

2.5. Beneficio de la Tecnología de objetos

La fiabilidad se puede mejorar debido a que cada objeto es una caja negra con respecto a otros objetos con quien debe comunicarse.

Las estructuras de datos y métodos se pueden refinar sin afectar a otras partes del sistema, es decir se puede hacer más eficiente un proceso codificado en una función sin problemas de compatibilidad con la aplicación que se sirve de la función, siempre y cuando los parámetros de entrada y el tipo de dato que arroja la función como resultado no cambien.

La productividad del desarrollador se puede mejorar, debido a que las clases de objetos se pueden hacer reutilizables, por otra parte esta productividad también aumenta, debido a que existe una asociación más natural del sistema y objetos del mundo real

Los beneficios genéricos de la tecnología orientada a objetos son:

- Reutilización de clases
- Sistemas más fiables
- Proceso de desarrollo más rápido
- Desarrollo más flexibles
- Modelos que reflejan mejor la realidad
- Biblioteca de clases comerciales disponibles
- Mejores relaciones con los clientes
- Mejora de la calidad del producto software terminado

2.6. La Programación Orientada a Objetos

La programación orientada a objetos (POO) es un método de implementación en el que los programas se organizan como colecciones cooperativas de objetos, cada uno de los cuales representan una instancia de una clase y cuyas clases son, todas

ellas, miembros de una jerarquía de clases unidas mediante relaciones de herencias.

En esta definición se pueden identificar tres palabras claves, las cuales son 1) Utilización de objetos, no algoritmos, 2) Un objeto es una instancia de una clase 3) Las clases están relacionadas con otras clases por medio de la herencia, un programa donde falte cualquiera de estos tres elementos no es un programa orientado a objetos. Específicamente, la programación sin herencia es explícitamente no orientada a objetos.

Según esta definición algunos lenguajes son orientados a objetos y otros no lo son como lo vimos en Taxonomía de los lenguajes orientados a objetos. Un cierto lenguaje es orientado a objetos si y sólo si satisface los siguientes requisitos:

- Soporta objetos
- Los objetos tienen un tipo asociado (Clases)
- Las clases pueden heredar atributos de las superclases (Herencia).

Un lenguaje que soporta la herencia significa que es posible realizar relaciones de estructura de clases “**Es un**” , por ejemplo, una rosa **es un** tipo de flor, y una flor **es un** tipo de planta. Si un lenguaje no ofrece soporte directo para la herencia entonces no es orientado a objetos. Los conceptos claves en la POO son: objetos, clases, herencia, mensajes, y polimorfismo.

2.6.1. Métodos y Mensajes

Un objeto es la combinación en una sola unidad de datos y funciones que operan sobre datos. Por consiguiente dentro de los datos residen los lenguajes de programación tradicionales, tales como números, arreglos, así como funciones y rutinas que operan sobre ellos.

Un programa orientado a objetos consiste en un número de objetos que se comunican unos con otros llamando a las funciones miembros que residen en el objeto y determinan como actúan según el mensaje recibido. Un mensaje es la acción que realiza un objeto. Mientras que un método es el procedimiento o función que se invoca para actuar sobre un objeto, éste especifica como se ejecuta un mensaje.

Pero ¿como sabe el objeto a que mensajes responder y a que mensajes no?. Para esto existe lo que se denomina protocolo del objeto, que son el conjunto de mensajes a los cuales puede responder un objeto.

Como ya se mencionó, la estructura interna de un objeto se encuentra oculta a los usuarios y programadores. Los mensajes que recibe el objeto son las únicas vías que lo conectan con el mundo exterior.

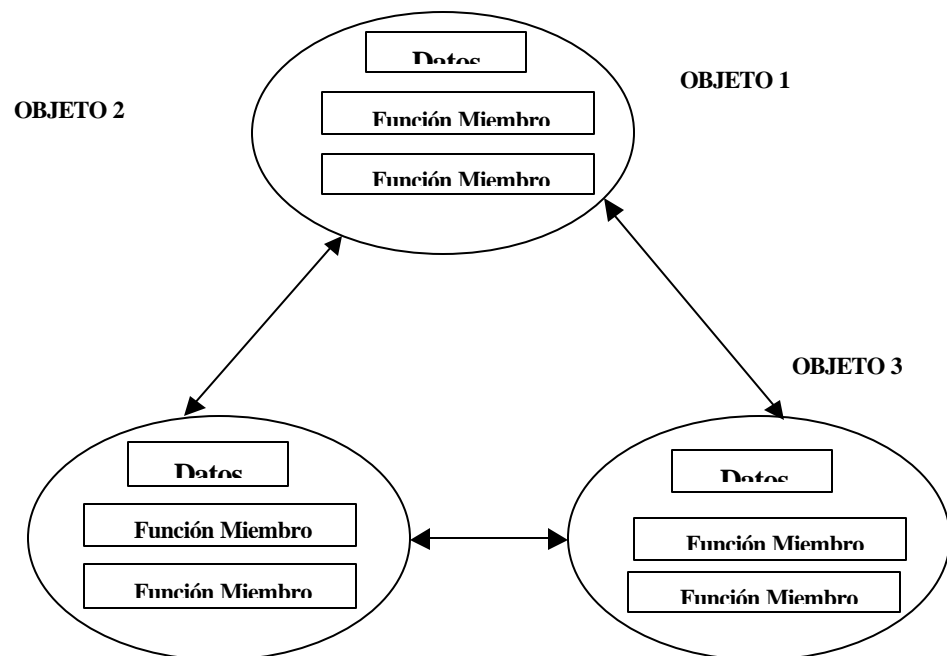


Figura 2.3. Mensajes entre objetos

2.6.2. Clases

Una clase es un conjunto de objetos que comparten una estructura común y un comportamiento común.

Los conceptos de clases y objetos están íntimamente ligados ya que no se puede hablar de un objeto sin poner atención a su clase. Sin embargo existen diferencias importantes entre ambos, una de la más importante es que un objeto representa algo concreto que existe en el tiempo y en el espacio, mientras que una clase es la esencia de un objeto (abstracción), así se puede hablar de los mamíferos que representan todas las características comunes de éstos, pero para particularizar en esta clase tendríamos que hablar de aquel mamífero o de ese mamífero.

Cada vez que se crea un objeto a partir de una clase estamos creando lo que se denomina una **instancia** de la clase, los objetos se crean cuando un mensaje de petición de creación se recibe por la clase base.

2.6.3. Identificación de Objetos

Si deseamos resolver un problema implementando un programa orientado a objetos, como es lógico lo primero que debemos de identificar son los objetos, es decir ¿Como definimos los objetos dentro del dominio de la definición del problema?.

Esta identificación se lo realiza mediante el análisis del problema, los objetos pueden caer dentro de las siguientes categorías.

- Cosas tangibles (Avión, casa, edificio, sillas).
- Roles o papeles jugados o representados por personas (gerente, cliente, ingeniero, médico).
- Organizaciones (empresas, división, equipo)
- Incidentes (evento, accidentes, muerte, vuelo)
- Interacciones (implica generalmente una transición o contrato y relacionan 2 o mas objetos del modelo: compras-comprador, vendedor- artículo, matrimonio-esposo, esposa, fecha de bodas)
- Especificaciones (muestran aplicaciones de inventario o fabricación: refrigeradora, nevera)
- Lugares (sala de embarque, muelle de carga)

Una vez identificados los objetos, se necesitan identificar las abstracciones de las características individuales de estos, a estas abstracciones se las conocen como atributos. Ejemplo

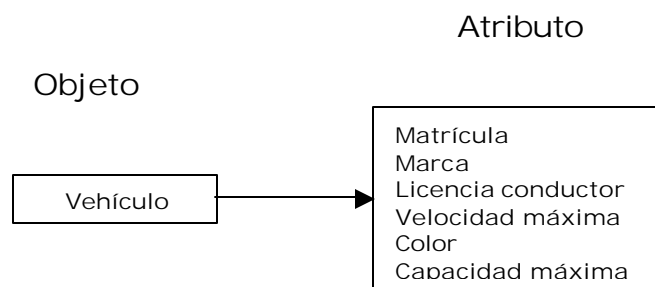


Figura 2.4. Atributos de un objeto

Existen operaciones que inciden directamente sobre los atributos, es preciso entonces reconocerlas, puesto que son las operaciones las que cambian el comportamiento de un objeto, es decir, cambian los valores de uno o más atributos contenidos en los objetos. Aunque existe un gran número de operaciones, generalmente se las pueden dividir en 3 grandes grupos.

- Operaciones que manipulan los datos de alguna forma específica (añadir, borrar, cambiar formato)
- Operaciones que realizan un cálculo o proceso
- Operaciones que comprueban un objeto frente a la ocurrencia de algún suceso de control

2.7. Comunicación entre objetos

Como se mencionó existe la posibilidad de que objetos diferentes mantengan una comunicación directa por medio del paso de mensajes, que no es más que la petición de un objeto llamado emisor a otro llamado receptor que invoque algún método en especial para que se efectúe una operación o proceso específico. Supongamos que en una aplicación pulsamos el botón "Salir", este envía un mensaje al objeto responsable de cerrar la aplicación. Si el mensaje es válido se invoca el método interno, entonces se cierra la aplicación, note que el objeto botón no le indica al objeto receptor como cerrar la aplicación, este maneja la operación cerrar por si mismo.

2.7.1. Activación de objetos

A los objetos se los puede acceder a través de su interfaz pública enviándole un mensaje, este es una petición de un objeto a otro al que le solicita ejecutar uno de sus métodos. Por convenio, el objeto que envía el pedido se denomina emisor y el objeto que recibe lo pedido se denomina receptor. Estructuralmente, un mensaje consta de tres partes:

- Identidad del receptor
- El método que se ha de ejecutar
- Información especial para realizar el método (parámetro)

Mientras un objeto no reciba ningún tipo de mensaje se dice que el objeto esta inactivo, si recibe un mensaje se hace activo.

2.7.2. Paso de Mensajes

Cuando un objeto recibe un mensaje éste identifica si el nombre del mensaje es considerado como uno de los mensajes que entiende, en otro caso el objeto no se activará, por ende no se invocará ningún método.

Los parámetros o argumentos que un mensaje puede tener son:

- Datos invocados por el método invocado
- Un mensaje propiamente dicho

2.8. Estructura interna de un objeto

La estructura interna de un objeto consta de 2 componentes básicos: atributos y métodos

2.8.1. Atributos

Los atributos describen el estado del objeto. Un atributo consta de 2 partes: un nombre de atributo, y un valor de atributo.

Los objetos simples pueden constar de tipos primitivos de datos, tales como enteros, carácter, boolean, reales, o tipos simples definidos por el usuario. Los objetos complejos pueden constar de pilas, conjuntos, listas arrays, etc. , e incluso estructuras recursivas de algunos o todos los elementos.

2.8.2. Métodos

Los métodos describen el comportamiento asociado a un objeto. Representan las acciones que pueden realizarse por un objeto o sobre un objeto. La ejecución de un método puede conducir a cambiar el estado del objeto.

Cada método tiene nombre y cuerpo que realiza la acción o comportamiento asociado con el nombre del método. En un LOO, el cuerpo de un método consta de un bloque de códigos que ejecuta la acción requerida. Todos los métodos que alteran o acceden a los datos de un objeto se definen dentro del objeto.

Así mismo como un objeto puede enviar un mensaje a otro para invocar algún método, también se pueden enviar mensajes locales, esto es mensajes que se envían de un método a otro dentro del objeto.

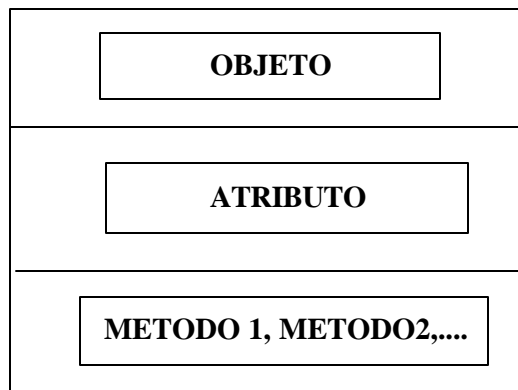


Figura 2.5. Mensajes internos

2.9. Herencias

Cuando un ser vivo capaz de procrear tiene un sucesor este hereda un número de características fisiológicas y anatómicas pero no todas, puesto que este tiene sus propias características que lo hacen único entre los demás. La idea de herencia en la programación orientada a objetos es muy similar a lo que sucede con los seres vivos, ésta característica da la capacidad a los objetos de utilizar estructuras de datos métodos previstos en sus antepasados o ascendientes. El objetivo final es la reutilización, es decir utilizar el código anteriormente desarrollado.

La herencia supone una clase base y una jerarquía de clases que contiene las clases derivadas de la clase base. Las clases

derivadas pueden heredar el código de la clase base, añadiendo su propio código especial.

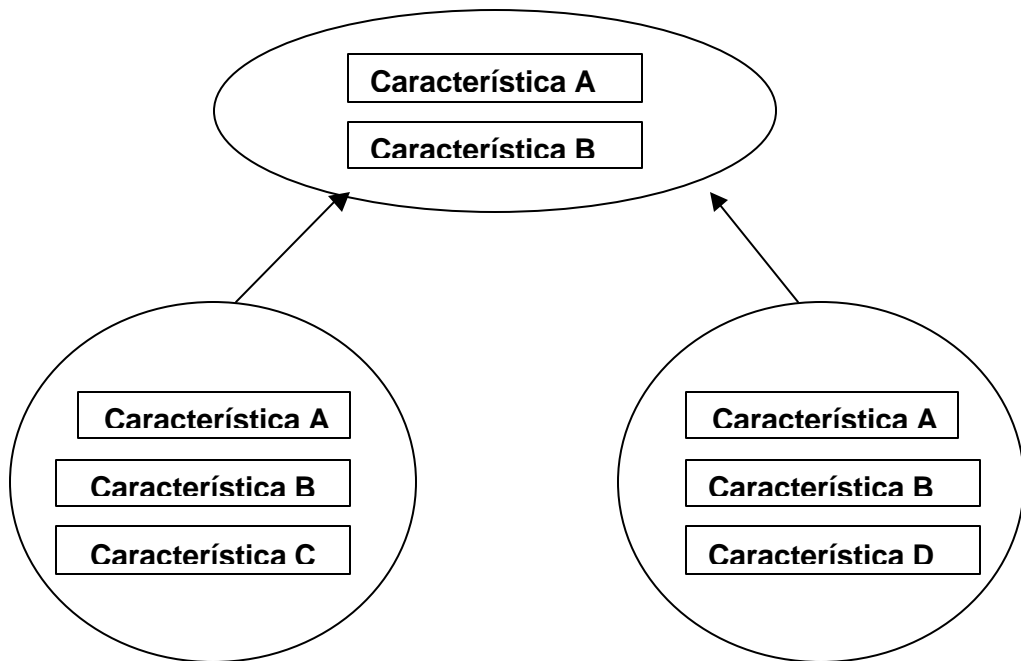


Figura 2.6. Herencia

Existen dos tipos de herencias comúnmente utilizadas en programación orientada a objetos, herencia simple y herencia múltiple.

2.9.1. Herencia simple

En este tipo de herencia se define que una subclase máximo tiene una superclase, este tipo de herencia permite que una clase herede las propiedades de su superclase en una cadena jerárquica

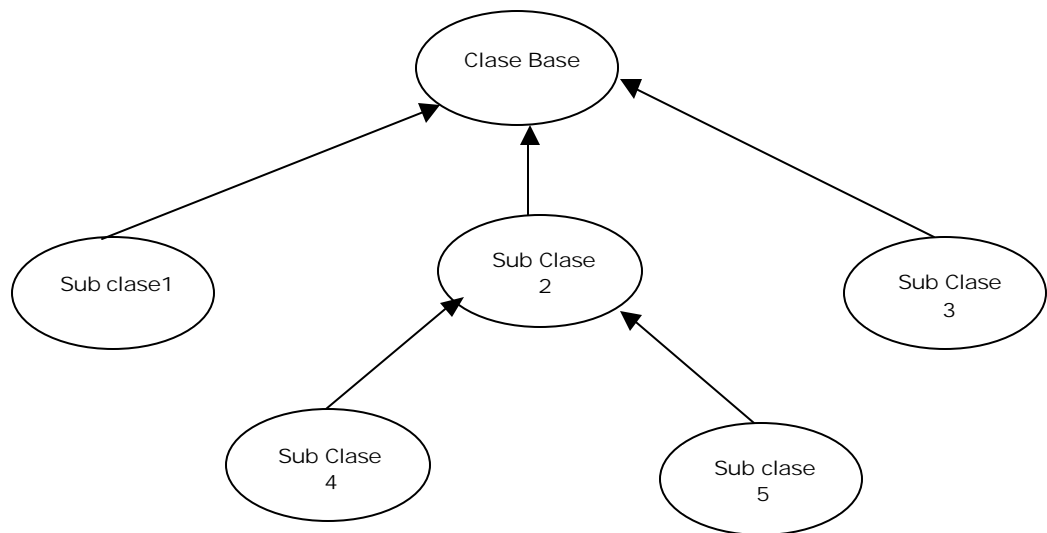


Figura 2.7 Herencia Simple

2.9.2. Herencia Múltiple

La herencia múltiple es una generalización de la herencia simple puesto que ésta consta de clases cada una de las cuales puede tener una o más superclases, por ejemplo la clase profesor

universitario puede heredar características de la superclase profesor y de la superclase investigador.

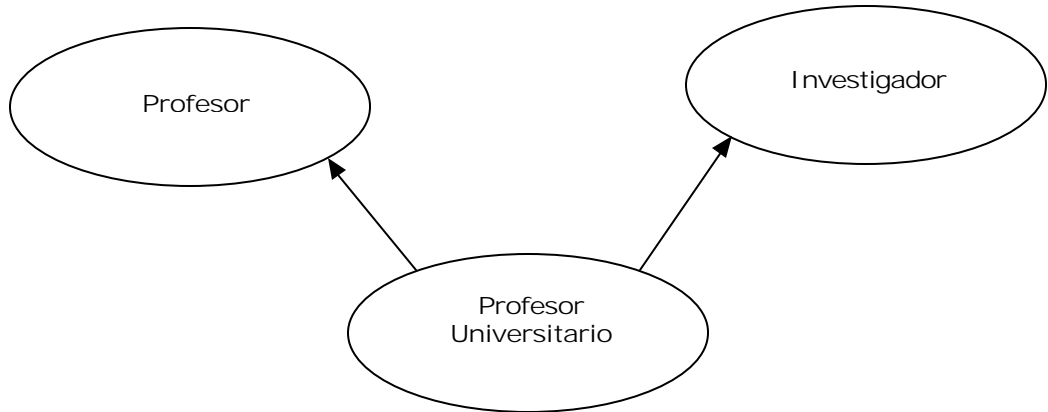


Figura 2.8. Herencia Múltiple

CAPITULO III

3. MARCO TEÓRICO DE LAS MATEMÁTICAS ACTUARIALES

Cuando una empresa empieza sus operaciones, lo primero que debe de tener bien definido son los objetivos que ésta persigue, entre ellos los financieros, a corto, mediano y largo plazo, que contribuyan con su misión y forjen su visión, para esto debe de imponerse metas continuas que ayuden a alcanzar los objetivos propuestos, pero la empresa se puede topar con eventos que afectarán directamente las metas de la compañía, por ejemplo la perdida total de los equipos de producción, el robo o daño de los vehículos de reparto de productos, la perdida de las instalaciones ya sea por eventos naturales como terremotos, inundación

etc. Todos estos eventos representarán un costo inesperado que la empresa debe cubrir si desea continuar con las actividades planificadas.

No existe empresa o persona que conozca con certeza lo que ocurrirá de aquí a mañana o de aquí seis meses, pero si podemos ser capaces de decir que tan probable es que ocurra un evento en un lapso de tiempo determinado, puesto que para cada evento hay asociado una probabilidad de ocurrencia, por lo tanto las empresas o personas necesitan un sistema para reducir el impacto financiero adverso ocasionado por sucesos aleatorios, que impiden que se concreten normalmente las expectativas futuras. A este sistema se lo denomina "Operación de Seguros" y las empresas encargadas de hacer efectivo este servicio son las empresas de seguros, todo esto bajo el sustento técnico de da las matemáticas actuariales, que no es más que la simbiosis de dos ciencias, la ingeniería económica y la estadística matemática.

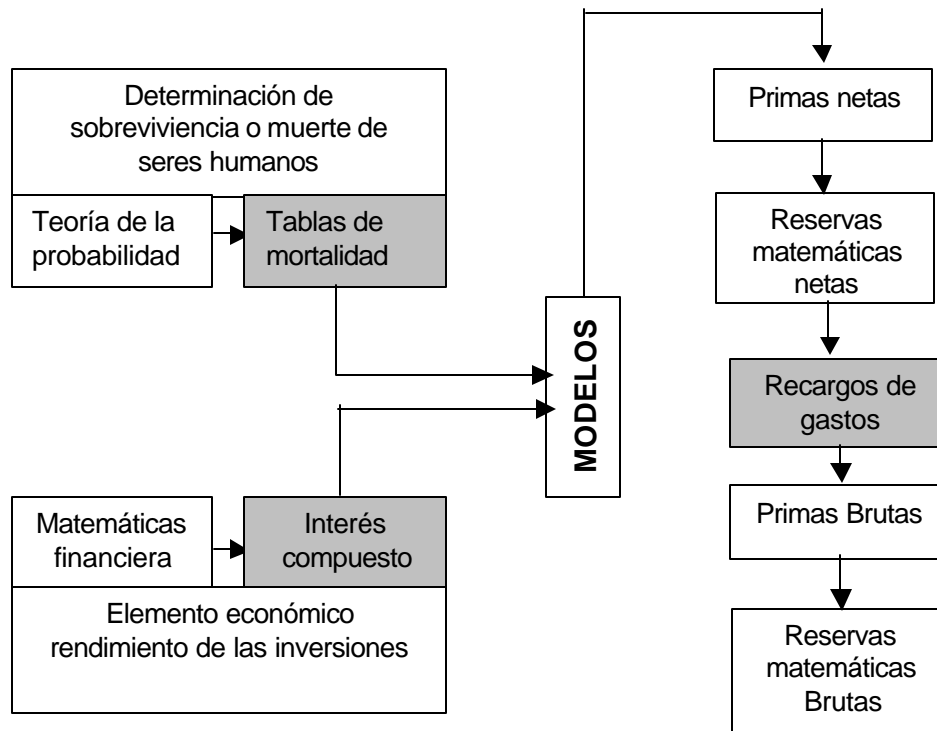


Figura 3.1. Estructura de un modelo actuarial.

La figura 3.1 muestra los componentes que forman un modelo actuarial, como se mencionó un modelo tiene dos partes una estadística y la parte financiera, la parte estadística básicamente es la encargada de dar fundamentos para el desarrollo de las tablas de mortalidad, donde se determine la supervivencia o muerte de los seres humanos. La parte financiera por el contrario se responsabiliza del cálculo de una tasa técnica acorde a las exigencias del mercado que permita a la compañía aseguradora obtener ganancias, y no perjudique a los clientes.

Con la tasa técnica y la tabla de mortalidad se puede desarrollar una gran cantidad de modelos matemáticos con características distintivas, de estos se obtienen las primas netas y las reservas matemáticas de las primas netas. Al añadirle los recargos de los gastos como los gastos administrativos o los gastos de producción se obtienen las primas brutas y las reservas matemáticas para estas primas, como se ve en la figura 3.1.

El principal problema del actuario o persona que especialista en la ciencia actuarial, es la de determinar con una visión a largo plazo, una prima adecuada para que la compañía de seguros de vida se mantenga en condiciones de poder pagar los siniestros. Además el nivel de prima debe de ser justificable a los ojos del asegurado.

Además del sustento financiero – estadístico, existen fundamentos económico que hacen que esta ciencia sea relevante para el desarrollo de un sector o de un país, como permitir que una empresa no deje de funcionar por elementos aleatorios, la consecuencia de esto es que la organización puede incursionar en actividades más riesgosas, que en otra situación no lo haría por las grandes pérdidas económicas que esto podría ocasionar.

3.1. Definiciones

Antes de desarrollar los aspectos que componen el marco teórico es importante definir el significado de algunas palabras y notaciones que se utilizarán con frecuencia en este capítulo.

Entes .- Se define como ente al grupo de individuos, colectividad, entidades, personas u objetos con características definidas.

Aseguradora.- Empresa encargada de emitir pólizas y desarrollar planes de seguros, para el beneficio del conglomerado.

Asegurado.- Persona u objeto al que se le asigna una póliza de seguros para cubrir la pérdida de alguna característica.

La notación que se utilizará para el acaecimiento de un evento cualquiera será (f/q) , esto puede ser el fallecimiento de una persona o la quiebra de una empresa. También denotaremos como (x) a un ente cuya edad es de x años.

3.2. Distribuciones de supervivencia

Supongamos que un ente tiene la edad 0, (0) (si el ente es una persona recién nacida, si es una empresa recién creada)

Entonces x es la variable aleatoria que denota la edad de ocurrencia de f/q del ente considerado.

La función de densidad de x se denomina función de pérdida y su función de distribución se define de la siguiente manera:

$$F(X) = P(x \leq X) \quad (3.1)$$

Que es la probabilidad, que un ente (0) fallezca o quiebre antes de cumplir x años, como se observa en la siguiente figura.

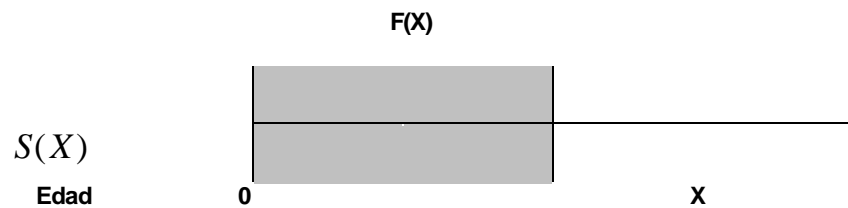


Figura 3.2. Rango de fallecimiento o quiebra de un ente, entre 0 y X años

Definamos entonces la función de supervivencia como el complemento de la fórmula 3.1

$$S(X) = 1 - F(X) = P(x > x) \quad (3.2)$$

Esta es la probabilidad que un ente cuya edad es cero sobreviva los x años, o f/q después de los x años. El siguiente diagrama ilustra este resultado.

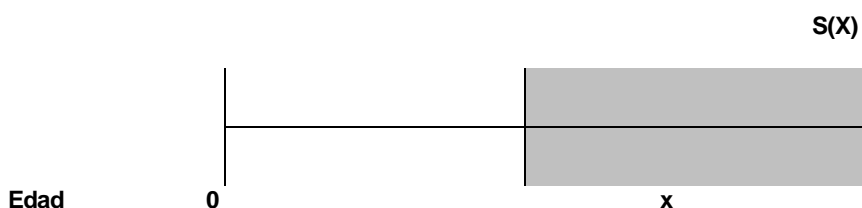


Figura 3.3. Rango de sobre vivencia de un ente entre 0 y X años de edad

Algunas propiedades de la función de supervivencia se describen a continuación:

Sea w la edad máxima de supervivencia, tenemos que.

$$\begin{aligned}
 a) \quad & S(x) > 0 \\
 b) \quad & S(0) = 1 \\
 c) \quad & S(w) = 0
 \end{aligned}
 \tag{3.3}$$

La primera propiedad es evidente puesto que $F(X) \leq 1$ para toda edad x . En la segunda propiedad tenemos que la probabilidad de que un ente sobreviva la edad 0 es 1, esto es cierto puesto que el ente está recién creado si es una empresa o recién nacido, si es una persona. Como w es el límite superior de supervivencia

entonces ningún ente podrá sobrevivir a la edad w así $F(w)=1$ por lo tanto $S(w)=0$ como nos indica el literal C.

Se define también la probabilidad de que un ente f/q dentro de un intervalo $[x-z]$ de las siguientes formas.

$$P(X < \mathbf{x} \leq Z) = \begin{cases} \mathbf{1)} & F(z) - F(x) \\ \mathbf{2)} & S(x) - S(z) \end{cases} \quad (3.4)$$

Cada uno de los casos se ilustra a continuación:

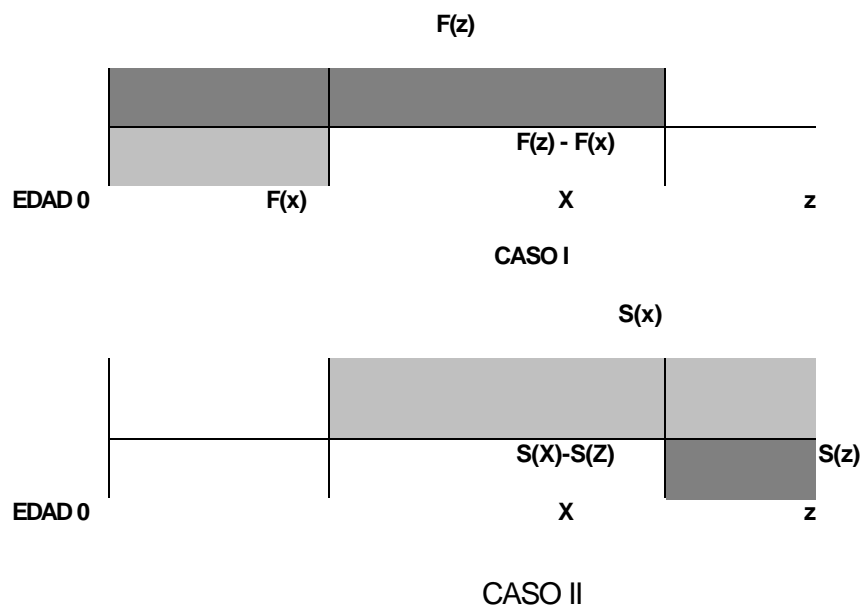


Figura 3.4. Intervalo de fallecimiento o quiebra de un ente

3.2.1. Tiempo futuro de supervivencia

Se va a obtener expresiones para las funciones del tiempo futuro de supervivencia y a suponer que el ente está condicionado a tener x años de edad.

El tiempo futuro de supervivencia es una variable aleatoria que representa el tiempo que le queda por vivir a un ente (x), a esta variable se la denota por T_x y se la define como.

$$T_x = \mathbf{x} - x \quad (3.5)$$

Observemos que T_x es una función lineal de la variable aleatoria \mathbf{x} así pues es factible obtener la densidad de T_x si se conociera la densidad de \mathbf{x} aplicando las técnicas estadísticas ya conocidas de funciones de variables aleatorias.

Ahora se define la función de distribución de T_x como

$${}_t q_x = P(T_x \leq t) \quad \forall t \geq 0 \quad (3.6)$$

La probabilidad de que un ente (x) , f/q dentro de el intervalo $[x, x+t]$ por lo tanto la probabilidad que un ente (x) sobreviva t años más es el complemento de la fórmula 3.6:

$${}_t p_x = 1 - {}_t q_x = P(T(x) > t) \quad \forall t \geq 0 \quad (3.7)$$

Nótese que para cada edad vamos a obtener funciones de distribución diferentes. Podemos obtener fácilmente algunas propiedades de estas funciones de tiempo futuro.

$${}_x p_o = P(\mathbf{x} - 0 > x) = P(\mathbf{x} > x) = S(X) \quad (3.8)$$

$${}_{x+t} p_o = {}_x p_o \cdot {}_t p_x \Rightarrow {}_t p_x = \frac{{}_{x+t} p_o}{{}_x p_o} = \frac{S(x+t)}{S(x)}$$

La expresión que representa la probabilidad de que un ente (x) sobreviva t años y fallezca dentro de los n años siguientes viene dado por:

$${}_{t/n} q_x = P(t < T \leq t+n) = {}_{t+n} q_x - {}_t q_x = {}_t p_x - {}_{t+n} p_x \quad (3.10)$$

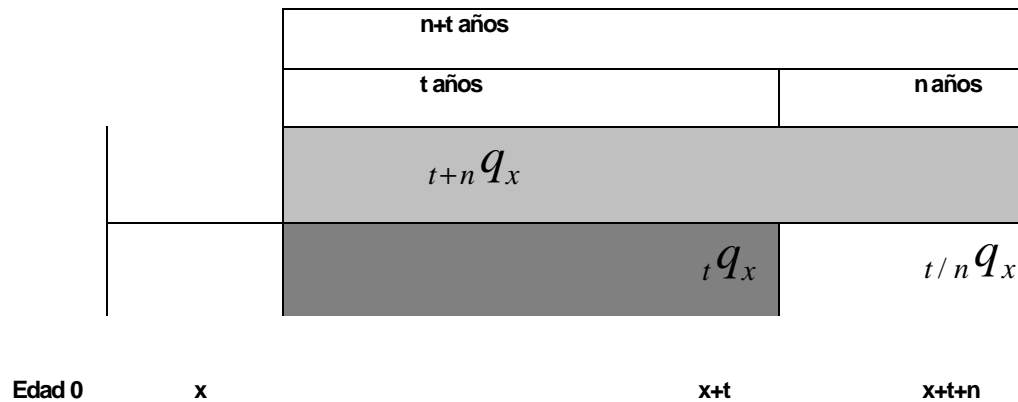


Figura 3.5. Probabilidad que un individuo (x) sobreviva t años más y fallezca después de t+n años

Aplicando los resultados obtenidos en 3.10 vamos a expresar la probabilidad antes mencionada en términos del producto de ${}_t p_x$ y ${}_t q_x$ de la siguiente forma.

$${}_{t/n}q_x = {}_t p_x \cdot {}_n q_{x+t} \quad (3.11)$$

3.2.2. Tiempo Futuro de Vida Abreviada

En la práctica generalmente la variable aleatoria, tiempo futuro de supervivencia asociada a modelos actuariales es discreta, esta

variable aleatoria la denotaremos por $K(x)$ y su función de probabilidad se la define como sigue.

$$P(K(X) = k) = P(k < T_x < k + 1) = {}_k p_x - {}_{k+1} p_x = {}_k p_x q_{x+k}$$

Para $k = 0, 1, 2, 3, \dots$

(3.12)

Esta función de probabilidad es un caso particular de las fórmulas 3.10 y 3.11 con $t = k$ y $n = 1$, por lo tanto a 3.12 se la puede escribir como.

$$P(K(X) = k) = P(k < T_x < k + 1) = {}_k q_x \quad \text{(3.13)}$$

Donde $K(x)$ representa el número de años completos que le queda por vivir a (x) .

3.2.3. Modelo de Supervivencia / quiebra

Las tablas de mortalidad traen tabulados valores de suma importancia para el cálculo de primas y reservas, como el número de personas que se espera sobrevivan a una edad x o el número esperado de personas que fallecen entre las edades x y $x+n$, entre otros valores.

Se Define al número de supervivientes a una edad x como la variable aleatoria $I(x)$ y l_0 como el número de recién nacidos, recién creadas o el tamaño de la cohorte del conglomerado de entes.

Para obtener la esperanza matemática de $I(x)$ (cuya estimación es la que aparece en una tabla de mortalidad) se define la siguiente variable bernulli como.

$$I_j(x) = \begin{cases} 0 & \text{Si el individuo } j \text{ sobrevive a la edad } x \\ 1 & \text{Caso contrario} \end{cases}$$

entonces $I(x) \approx B(S(x))$

Con esto tenemos que $I(x) = \sum_{j=1}^{l_0} I_j(x)$

Por lo tanto el número de personas que se espera sobrevivan a la edad x está dado por.

$$l_x = \sum_{j=1}^{l_0} E(I_j(x)) = \sum_{j=1}^{l_0} S(x) = l_0 S(x) \quad (3.14)$$

Observe que si obtenemos los valores estimados de l_x los valores estimados de $S(x)$ son viables de calcular.

Como ya se mencionó las tablas de mortalidad traen tabulados también el número de entes esperados que f/q dentro de un intervalo de edad $[x, x+n]$, esta variable esta representada por ${}_n\partial_x$ y su valor esperado por ${}_n d_x = E({}_n\partial_x)$. Se deduce por la fórmula 3.14 que

$${}_n d_x = l_o(S(x) - S(x+n)) = l_x - l_{x+n} \quad (3.15)$$

Teniendo como base lo desarrollado hasta aquí se demostrará que

$$\begin{aligned} \text{a)} \quad {}_t q_x &= \frac{{}_t d_x}{l_x} \\ \text{b)} \quad {}_t p_x &= \frac{l_{x+t}}{l_x} \end{aligned} \quad (3.16)$$

Para demostrar el literal (a) la parte derecha de la igualdad se la puede expresar como.

$$\begin{aligned} \frac{{}_t d_x}{l_x} &= \frac{l_x - l_{x+t}}{l_x} = 1 - \frac{l_{x+t}}{l_x} = 1 - \frac{l_0 S(x+t)}{l_0 S(x)} = 1 - \frac{S(x+t)}{S(x)} & (3.17) \\ &= 1 - {}_t p_x = {}_t q_x \end{aligned}$$

Al demostrar el literal (a) implícitamente el literal (b) queda demostrado.

Las expresiones obtenidas en 3.16 son importantes ya que si en una tabla de mortalidad se tienen los valores de ${}_n d_x$ o l_x , es sencillo estimar la función de probabilidad de los $K(x)$ para $x = 0, 1, 2, \dots$

3.2.4. Tanto instantáneo de f/q

La probabilidad de que un ente (x), f/q dentro de 1 año esto es ($x+1$), viene dado por ${}_1 q_x$ (representado también por q_x) es evidente que la intensidad de f/q varía conforme el ente avanza en el tiempo, entonces sería interesante obtener la variación de esta intensidad considerando un aumento en x muy pequeño ($x + \Delta x$).

Por ejemplo el tanto instantáneo de mortalidad para un ente de 23 años de edad mide la probabilidad que este f/q dentro de 5

minutos, o si hablamos de un delta más pequeño, que este f/q en 5 segundos o microsegundos.

Entonces tenemos que la probabilidad condicional que un ente f/q en el intervalo $(x, x + \Delta x)$ dado que ya sobrevivió x años es igual a:

$$P(x < \mathbf{x} \leq x + \Delta x / \mathbf{x} > x) = \frac{F(x + \Delta x) - F(x)}{1 - F(x)} \quad (3.18)$$

Considerando la definición de la derivada de una función continua, tenemos que la fórmula 3.18 también se la puede expresa como:

$$\frac{F(x + \Delta x) - F(x)}{1 - F(x)} \approx \frac{F'(x)\Delta x}{1 - F(x)} = \frac{f(x)\Delta x}{1 - F(x)} \quad (3.19)$$

$$u_x = \frac{f(x)}{1 - F(x)} \quad (3.20)$$

De donde se tiene que 3.20 es el tanto instantáneo de f/q o tasa de mortalidad y se denota por m_x esta tasa es de suma importancia en las matemáticas actuariales, puesto que de aquí se derivan todas las funciones y valores esperados que se han

considerado tales como, l_x , $S(x)$, ${}_tP_x$ entre otros, como se demostrará a continuación.

Se sabe por la fórmula 3.2 que

$$S(x) = [1 - F(x)]$$

$$\frac{\partial S(x)}{\partial x} = -f(x)$$

Por lo tanto

$$\mathbf{m}_x = \frac{f(x)}{1 - F(x)} = -\frac{S'(x)}{S(x)} \quad (3.21)$$

Se sabe también que la derivada de un logaritmo cuyo argumento es una función continua es:

$$\frac{\partial}{\partial x} \ln(f(x)) = \frac{f'(x)}{f(x)} \quad (3.22)$$

De las fórmulas 3.21 y 3.22 se puede deducir que la tasa de mortalidad también es igual a:

$$\mathbf{m}_x = -\frac{S'(x)}{S(x)} = -\frac{\partial}{\partial x} \ln(S(x)) \quad (3.23)$$

Si multiplicamos a ambos lados de la ecuación por el diferencial se tiene:

$$\mathbf{m}_y \partial y = \partial \ln(S(y)) \quad (3.24)$$

Así se obtiene una ecuación diferencial, posteriormente tomaremos la integral en ambos lados de la igualdad y se evaluará ésta en el intervalo $[x, x+n]$.

$$\int_x^{x+n} \mathbf{m}_y \partial y = \int_x^{x+n} \partial \ln(S(y)) \partial y = \ln(S(x+n)) - \ln(S(x)) = \ln\left(\frac{S(x+n)}{S(x)}\right)$$

$$\int_x^{x+n} \mathbf{m}_y \partial y = \ln({}_n P_x) \quad (3.25)$$

Despejando ${}_n P_x$ obtenemos

$${}_n P_x = e^{\int_x^{x+n} \mathbf{m}_y \partial y} \quad (3.26)$$

Finalmente consideremos el siguiente cambio de variable $t = y - x$

$${}_n P_x = e^{-\int_0^n m_{t+x} dt} \quad (3.27)$$

Haciendo uso de 3.27, estamos en capacidad de encontrar una expresión para la función de supervivencia en términos del tanto instantáneo de mortalidad, de la siguiente forma.

$$S(x) = {}_x P_0 = e^{-\int_0^x m_t dt} \quad (3.28)$$

La tasa de mortalidad posee características exclusivas, que permiten identificarla como tal, por ejemplo la tasa de mortalidad es siempre no - acotada esto es $\int_0^{\infty} u_t dt = \infty$ para demostrar esta afirmación vamos a utilizar el resultado en 3.26.

Se Tiene que $\ln {}_n P_x = -\int_x^{x+n} u_t dt$ además

$$\lim_{n \rightarrow \infty} (\ln {}_n P_x) = \ln(\lim_{n \rightarrow \infty} {}_n P_x) = \ln(0) = -\infty \Rightarrow \lim_{n \rightarrow \infty} \int_x^{x+n} u_t dt = \infty$$

Por lo tanto la tasa de mortalidad no es acotada. Así mismo utilizando el resultado en 3.28 el número de personas que se espera sobreviva a la edad x es:

$$l_x = l_0 S(x)$$

$$l_x = l_0 e^{-\int_0^x \mu_t dt} \quad (3.29)$$

Obviamente se puede obtener expresiones para ${}_t q_x$ puesto que es el complemento de ${}_t p_x$, lo que no es tan obvio es obtener la función de densidad de T_x en términos de la tasa instantánea de mortalidad, este proceso lo vamos a considerar a continuación.

Se sabe que la función acumulada del tiempo futuro de supervivencia viene dado por

$$p(T_x < t) = {}_t q_x$$

Y que

$${}_t q_x = 1 - \frac{S(x+t)}{S(x)}$$

Entonces la función de densidad de T_x denotada como $f_x(t)$ es:

$$f_T(x) = \frac{\partial}{\partial t} \left(\frac{S(X) - S(X+T)}{S(X)} \right) = -\frac{1}{S(X)} \left(\frac{\partial}{\partial t} S(X+T) \right) \quad (3.30)$$

Multiplicando a la última expresión de 3.30 por $\frac{S(X+T)}{S(X+T)}$ se tiene que.

$$f_T(x) = - \left(\frac{S'(x+t)}{S(x+t)} \right) \left(\frac{S(x+t)}{S(x)} \right)$$

$$f_T(x) = \mathbf{m}_{x+t} P_x \quad (3.31)$$

El último resultado de 3.31 se puede interpretar como la probabilidad de que un ente (x) sobreviva a x+t y f/q en $(x+t+\Delta_{x+t})$. Al integrar en un intervalo de tiempo lo que sucede es que t varía en incrementos muy pequeños, así consideramos el acaecimiento de f/q sobre todos los puntos de la recta real del tiempo.

Por lo tanto

$${}_n q_x = \int_0^n \mathbf{m}_{x+t} P_x \partial t$$

$${}_{n/m} q_x = \int_0^{n+m} \mathbf{m}_{x+t} P_x \partial t \quad (3.32)$$

3.2.5. Leyes de Mortalidad Teóricas

Los modelos matemáticos han sido desarrollados principalmente para simular de la manera más precisa el entorno en que nos desenvolvemos, es interesante entonces que la tasa de mortalidad de un conglomerado siga una ley analítica, existen diferentes razones que sustentan éste interés, una de las principales es la filosófica, el cual nos dice que algunos fenómenos estudiados en la física pueden ser explicados en forma eficiente por simples fórmulas, utilizando argumentos biológicos, algunos autores han supuesto que la supervivencia humana es gobernada por una simple ley, adicionalmente algunas leyes analíticas tienen elegantes propiedades. Entonces por estas y otras razones sería muy interesante conocer si la tasa de mortalidad de alguna población esta modelada por alguna ley analítica.

Actualmente con el avance de la computación en aspectos como la velocidad y eficiencia se puede analizar con más detalles y de manera más precisa las propiedades de estas leyes, y con la ayuda de algunos métodos numéricos y estadísticos estimar la ley de mortalidad que mejor se ajusta a un conglomerado.

En el transcurso del tiempo varios personajes han aportado su esfuerzo y conocimiento formulando hipótesis para el desarrollo y análisis de leyes de mortalidad. Entre uno de ellos tenemos a Gompertz que en 1825 supuso que el poder de la muerte decrece a un tanto proporcional. Entonces puesto que u_x es una medida de la propensión a la muerte de (X) , Gompertz uso el recíproco $\frac{1}{u_x}$ para medir la resistencia a la muerte.

Su hipótesis se puede expresar matemáticamente de la siguiente forma.

$$\frac{\partial}{\partial x} \left(\frac{1}{u_x} \right) = -h \left(\frac{1}{u_x} \right)$$

Donde h es la constante de proporcionalidad.

Desarrollando la ecuación diferencial para encontrar u_x se va a tener:

$$\frac{\partial}{\partial x} (u_x)^{-1} = -h \left(\frac{1}{u_x} \right) \equiv - \left(\frac{1}{u_x} \right)^2 \frac{\partial}{\partial x} (u_x) = -h \left(\frac{1}{u_x} \right)$$

Multiplicando la última expresión por u_x a ambos lados de la igualdad tenemos:

$$\left(\frac{1}{u_x}\right) \frac{\partial}{\partial x}(u_x) = h \equiv \frac{\partial}{\partial x} \ln(u_x) = h \equiv \partial \ln(u_x) = h \partial x$$

Integrando la última igualdad se obtiene

$$\ln(u_x) = xh + k \equiv u_x = e^{xh + k} \tag{3.33}$$

Consideremos ahora los siguientes cambios de variables

$$c = e^h$$

$$B = e^k$$

Para tener finalmente

$$u_x = Bc^x \tag{3.34}$$

Para $B > 0, c > 1, x > 0$

Esto significa que la hipótesis de Gompertz equivale a decir que la fuerza de mortalidad crece en progresión geométrica. De aquí determinaremos la expresión para $S(X)$, el cual viene dado por.

$$S(x) = e^{-\frac{B}{\ln c} (c^x - 1)} \quad (3.35)$$

Makeman en 1860 combinó dos causas: una aleatoria, la previa disposición de (X) a la muerte o bien deterioro; la otra la inhabilidad creciente a resistir la destrucción, por lo tanto la ley de Makeman se establece añadiendo una constante a la de Gompertz:

$$u_x = Bc^x + A \quad \text{Para } A > -B, c > 1, x > 0 \quad (3.36)$$

A partir de esta ley también lograremos obtener una expresión para la función de supervivencia.

$$S(x) = e^{-Ax - \frac{B}{\ln c} (c^x - 1)} \quad (3.37)$$

Otra ley de mortalidad conocida es la ley de Weibull desarrollada más recientemente en 1939, su expresión matemática es la siguiente:

$$\text{Para } x > 0, n > 0 \quad u_x = kX^n \quad (3.38)$$

La respectiva función de supervivencia es

$$S(X) = e^{-ux^{n+1}} \quad (3.39)$$

3.2.6. Esperanza de vida abreviada y completa

La esperanza de vida de un ente de edad (x) viene representado de dos formas e_x y e_x^o para denotar la esperanza de vida abreviada y completa respectivamente, la diferencia entre ambas radica que, en la esperanza de vida abreviada toma en consideración los años completos de vida futura, mientras que en la esperanza de vida completa toma en consideración todo el campo continuo, esto es fracciones de años.

Entonces se tiene que la esperanza de vida abreviada viene dada por:

$$e_x = \sum_{k=0}^{w-x-1} k {}_k p_x q_{x+k} = \sum_{k=0}^{w-x-1} {}_{k+1} p_x \quad (3.40)$$

La esperanza de vida completa se la calcula de la siguiente forma:

$$e_x^o = \int_0^{w-x} {}_t p_x \mathbf{m}_{x+t} \partial t \quad (3.41)$$

Si esta última expresión la integramos por partes tenemos:

$$e_x^o = \int_0^{w-x} {}_t p_x \partial t \quad (3.42)$$

Las dos esperanzas están relacionadas por la siguiente expresión:

$$e_x^o \approx e_x + \frac{1}{2} \quad (3.43)$$

Asimismo, se puede definir la esperanza de vida temporal de la siguiente manera. Para el caso de la esperanza de vida abreviada

$$e_{x:n} = \sum_{t=1}^n {}_t p_x \partial x \quad (3.44)$$

Para la esperanza de vida completa

$$e_{x:n}^o = \int_0^n {}_t p_x \partial x \quad (3.45)$$

3.2.7. Número total esperado de años de supervivencia

Considerando ahora un determinado colectivo de personas o sector de empresas constituido inicialmente por l_0 elementos, se denota por L_x “el número total esperado de años de supervivencia” vividos por los elementos del colectivo y lo expresaremos.

$$L_x = \int_0^1 t l_{x+t} m_{x+t} dt + l_{x+t} \quad (3.46)$$

Integrando por partes obtenemos:

$$L_x = \int_0^1 l_{x+t} \partial t \quad (3.47)$$

3.2.8. Tanto central de f/q a la edad (x)

$$m_x = \frac{l_x - l_{x+1}}{L_x} \quad (3.48)$$

Se denota por T_x el número total de años vividos desde x por el grupo de supervivientes procedentes de un grupo inicial l_0 elementos y se puede expresar de la siguientes formas.

$$T_x = \int_0^{\infty} t l_{x+t} m_{x+t} \partial t = - \int_0^{\infty} t dl_{x+t} = \int_0^{\infty} l_{x+t} \partial t \quad (3.49)$$

Por otra parte tenemos que:

$$l_{x+t} m_{x+t} = l_{xt} p_x m_{x+t} \quad (3.50)$$

El número medio de años de supervivencia futura de los l_x sobrevivientes del grupo a la edad x sería.

$$\frac{T_x}{l_x} = \frac{\int_0^{\infty} l_{x+t} \partial t}{l_x} = \int_0^{\infty} {}_t p_x \partial t = e_x^o \quad (3.51)$$

3.3. Modelos para las operaciones de seguros

En esta sección se incluirá la parte financiera, el cual servirá para desarrollar modelos los cuales brindarán una forma de reducir el impacto financiero asociado al evento aleatorio de f/q .

Un modelo esta compuesto principalmente de dos componentes el uno aleatorio y el otro financiero. Un modelo de seguros tiene asociado algunas características definidas que lo distinguen en forma única de los restante modelos, por ejemplo la indemnización que la compañía aseguradora tiene la obligación de cancelar si es que sucede el evento f/q , ésta puede ser constante, incremental, geométrico, etc. Otra característica importante es el momento cuando se cancela la indemnización pactada, ésta puede ser de dos formas al momento de f/q o al final del año de f/q , otra característica es el riesgo asociado al modelo etc.

3.3.1. Tasa de Interés Continua

La tasa de interés utilizada para los modelos continuos que se desarrollarán es la tasa continua esto es que el capital invertido se capitaliza de manera instantánea.

Dejando por un momento la parte probabilística y ahondando más en la financiera vamos a calcular una expresión para la tasa continua, para esto se va a suponer que ∂ es la tasa nominal capitalizable en forma continua y que i es la tasa efectiva compuesta continuamente, entonces tenemos la siguiente tautología:

$$(1+i) = \left(1 + \frac{\partial}{m}\right)^m \quad (3.52)$$

Donde m es el número de periodos de capitalización.

Un ejemplo para ver la aplicación de la expresión 3.52 sería el siguiente, suponga que una compañía de crédito anuncia una tasa de interés mensual del 2% se requiere la tasa de interés efectiva anual.

Tenemos entonces que $\partial = 2 \times 12 = 24$ es la tasa nominal anual compuesta mensualmente, el período de capitalización $m = 12$, la tasa efectiva mensual es.

$$i = \left(1 + \frac{\partial}{m}\right)^m - 1 = \left(1 + \frac{0.24}{12}\right)^{12} - 1 = 26.82\%$$

Nótese que en este ejemplo $m = 12$ pero que sucede cuando $m \rightarrow \infty$ esto es, cuando el periodo de capitalización es demasiado corto y el valor m número de periodos de capitalización se incrementa en forma infinita, entonces se tendrá que calcular el siguiente límite.

$$i = \lim_{m \rightarrow \infty} \left(1 + \frac{\partial}{m}\right)^m - 1 \quad (3.53)$$

y recordando que

$$i = \lim_{h \rightarrow \infty} \left(1 + \frac{1}{h}\right)^h = e = 2.71828\dots$$

se tiene finalmente

$$i = \lim_{h \rightarrow \infty} \left(1 + \frac{1}{h}\right)^{\partial h} - 1 = e^{\partial} - 1 \quad (3.54)$$

Como se menciono que (i) es la tasa efectiva anual capitalizable continuamente. Al despejar ∂ de la ecuación 3.54 tenemos que

$$\partial = \ln(1+i) \equiv -\partial = \ln\left(\frac{1}{1+i}\right) = \ln v \quad (3.55)$$

siendo $v = \left(\frac{1}{1+i}\right)$

de aquí obtenemos que

$$v = e^{-\partial} \quad (3.56)$$

De manera general podemos definir que

$$v_t = \left(\frac{1}{1+i}\right)^t = e^{-\int_0^t \partial_s ds} \quad (3.57)$$

Notemos que la expresión 3.56 es un caso particular de la expresión 3.57 cuando $t = 1$ y ∂_s una constante $\partial_s = \partial$. De aquí en adelante vamos a asumir esta situación a menos que se señale lo contrario.

3.3.2. Valor actuarial

Un individuo (x) contrata una operación de seguro de vida entera, cuya indemnización es de 1 unidad monetaria al momento en que este fallece, suponga ahora que esto ocurre t años después de haber adquirido el seguro, entonces la aseguradora en el momento de haber emitido la póliza tuvo que haberse cubierto con por lo menos $v_t = \left(\frac{1}{1+i}\right)^t$ u.m para así después de t años cancelar lo pactado. Pero recordemos que t es una variable aleatoria y la aseguradora no conoce exactamente el periodo de capitalización (esto es el tiempo futuro de supervivencia), para este caso la aseguradora se cubre con el valor promedio de los valores actuales, siguiendo este método va a ver ocasiones en que la aseguradora va a perder si el evento de f/q ocurre en un periodo de capitalización más largo que el promedio, y va a ganar si ocurre lo contrario.

Vamos asociar al momento de la pérdida de la característica (x) como T el cual es el número de años futuros de sobre vivencia de un ente de edad (X) entonces el valor financiero actual se define como v^T siendo $v = \frac{1}{1+i}$ donde i es la tasa técnica de interés. Entonces el valor actual no es más que el valor presente de una inversión (en este caso el monto a indemnizar), puesto a T años.

Como T es una variable aleatoria entonces v^T también lo es, de aquí tenemos que el valor actuarial es el valor esperado de v^T . Esto es:

Valor Actuarial = $E(v^T) = \int_0^{\infty} v^t dF_x(t)$ donde $F_x(t)$ es la función de distribución de T_x

Es posible también definir el valor actuarial en términos de la función de densidad expresada en 3.31 de la siguiente forma.

$$E(v^T) = \int_0^{\infty} v^t f_x(t) dt = \int_0^{\infty} v^t {}_tP_x u_{x-t} dt \quad (3.58)$$

El riesgo de un modelo de seguro se lo define como la varianza del valor actual así tenemos que la varianza de v^T es:

$$VAR(v^T) = E(v^{2T}) - E^2(v^T) = \int_0^{\infty} v^{2t} {}_tP_x u_{x+t} - \left(\int_0^{\infty} v^t {}_tP_x u_{x+t} \right)^2 \quad (3.59)$$

Las expresiones 3.58 y 3.59 se pueden escribir en términos de la fuerza de interés, pero antes es importante enunciar el siguiente teorema.

Teorema.- Para un seguro de vida sobre (X), se define la fuerza de interés en el tiempo t como ∂_t y los beneficios y función de descuento como b_t y v_t respectivamente. Si $(b_t)^j = b_t$ para todo t, entonces el $E[(b_t v_t)^j]$ calculado a la fuerza de interés ∂_t es igual al $E[(b_t v_t)]$ calculado a la fuerza de interés $j\partial_t$ para $j > 0$.

Del teorema se deriva entonces que los momentos del valor actual son:

$$\text{Sea } Z = v^T \text{ entonces } E(Z^k) = \int_0^{\infty} v^{tk} f_x(t) = \int_0^{\infty} e^{-\partial k t} f_x(t) \quad (3.60)$$

De aquí se deduce que las expresiones en 3.59 y 3.60 se pueden escribir como:

$$E(Z) = \int_0^{\infty} v^t {}_tP_x u_{x+t} dt = \int_0^{\infty} e^{-\partial t} {}_tP_x u_{x+t} dt \quad (3.61)$$

$$VAR(Z) = E(Z^2) - E^2(Z) = \int_0^{\infty} e^{-2\partial t} {}_tP_x u_{x+t} - \left(\int_0^{\infty} e^{-\partial t} {}_tP_x u_{x+t} \right)^2$$

Para calcular el valor actuarial y el riesgo con las fórmulas 3.60 y 3.61 es necesario por lo menos conocer la ley analítica que modele la fuerza o tasa de mortalidad, pero en la vida real esto no es común por lo que se debe encontrar una vía práctica para el cálculo de los valores actuariales, es aquí precisamente donde cobra importancia las tablas de mortalidad, las integrales se convierten en sumatorias y la función de densidad del tiempo futuro de supervivencia en la función de probabilidad de vida abreviada, entonces tenemos que.

$$E(Z) = \sum_{k=0}^{w-x} v^k {}_kP_x \cdot q_{x+k} = \sum_{k=0}^{w-x} v^k q_x$$

$$VAR(Z) = E(Z^2) - E^2(Z) = \sum_{k=0}^{w-x} v^{2k} q_x - \left(\sum_{k=0}^{w-x} v^k q_x \right)^2 \quad (3.62)$$

Para calcular el riesgo es necesario encontrar la tasa marginal al doble de la fuerza de interés $v^{2t} = e^{-2\delta t}$ entonces tenemos:

$$\begin{aligned}\delta &= \ln(1+i) \\ 2\delta &= \ln(1+j) \equiv 2\ln(1+i) = \ln(1+j) \Rightarrow j = i^2 + 2i\end{aligned}\quad (3.63)$$

Reemplazando la tasa original (i) se podrá calcular el segundo momento, como si se estuviera encontrando el valor actuarial del modelo, pero a una tasa marginal (j) de interés.

Tenemos ya una idea general de lo que es el valor actuarial y el riesgo, considerando fundamentalmente a v^T como una variable aleatoria, ahora se va a definir varios modelos de seguros, siguiendo una metodología y análisis muy parecido al hasta aquí presentado.

3.3.3. Valores Actuariales de modelos de seguros pagaderos al final y al momento del año f/q

Los modelos de seguros consideran dos tipos de posibilidades en el pago de la indemnización en el caso que suceda el f/q, esto es la

compañía aseguradora puede cancelar el monto pactado en el momento en que ocurre f/q o al final del año.

Para ambos casos el procedimiento de cálculo es similar pero los elementos a utilizar son distintos, por ejemplo para el primer caso se requiere conocer la función de densidad del tiempo futuro de supervivencia o la ley analítica de la tasa de mortalidad, algo no muy común como ya se mencionó, para el segundo caso se requerirá solamente las probabilidades anuales de fallecimientos, puesto que de estos se generan los restantes parámetros estimados que aparecen en una tabla de mortalidad, así como los símbolos de conmutación.

Los símbolos de conmutación son valores elaborados para cada edad con el fin de simplificar los cálculos, estos consideran la parte probabilística y financiera, son los siguientes.

$$\begin{aligned}
 C_x &= v^{x+1} d_x & M_x &= \sum_{t=0}^{\infty} C_{x+t} & N_x &= \sum_{t=0}^{\infty} M_{x+t} \\
 D_x &= v^x l_x & R_x &= \sum_{t=0}^{\infty} D_{x+t} & S_x &= \sum_{t=0}^{\infty} R_{x+t}
 \end{aligned}
 \tag{3.64}$$

Para todos los modelos vamos a suponer una indemnización de 1 u.m. Se acoge esta suposición por la característica de linealidad de la esperanza matemática, así si deseamos encontrar el valor actuarial de un seguro cuya indemnización es de b u.m simplemente se multiplicara este valor por el valor actuarial calculado a 1 u.m. Tenemos entonces los siguientes modelos de seguros.

3.3.3.1. Modelos sin años de diferimiento

Modelo 1

Seguro cuya prestación consiste en el pago del capital unitario de un ente de edad (x), siempre que su f/q acaezca pasado k años y precisamente en el año siguiente.

Pagaderos al momento de f/q

El Valor Actual es

$${}_{k/1}\bar{A}_x \begin{cases} v^t & t > k \text{ y } t < k+1 \\ 0 & \text{Para el resto} \end{cases}$$

Se define el valor actuarial y el riesgo como

$$\begin{aligned}
 E({}_{k/1}\overline{\mathbf{x}}_x) &= {}_{k/1}\overline{A}_x = \int_0^{k+1} v^t {}_tP_x u_{x+t} dt \\
 \text{Var}({}_{k/1}\mathbf{x}_x) &= {}_{k/1}{}^2\overline{A}_x - ({}_{k/1}\overline{A}_x)^2
 \end{aligned}
 \tag{3.65}$$

Donde ${}_{k/1}{}^2\overline{A}_x$ es el segundo momento de ${}_{k/1}\overline{\mathbf{x}}_x$

Pagaderos al final del año de f/q

El valor actual es cero si el ente de edad (x) f/q dentro de k años], es v^{t+1} si el ente f/q entre $[x + k, x + k + 1]$ y cero si el ente sobrevive la edad $x + k + 1$. Entonces el valor actual financiero viene dado por.

$${}_{k/1}\mathbf{x}_x \left\{ \begin{array}{ll} 0 & \text{con probabilidad } {}_kq_x \\ v^{k+1} & \text{con probabilidad } {}_kq_x \\ 0 & \text{con probabilidad } {}_{k+1}P_x \end{array} \right.$$

El valor actuarial viene dado por:

$${}_{k/1}A_x = E({}_{k/1}\mathbf{X}_x) = v^{k+1} {}_{k/1}q_x = v^{k+1} \left(\frac{d_{x+k}}{l_x} \right) = \frac{v^{x+k+1} d_{x+k}}{v^x l_x} = \frac{C_{x+k}}{D_x} \quad (3.66)$$

Aquí comenzamos a utilizar los símbolos de conmutación los cuales vienen tabulados en las tablas financiera – actuarial

Modelo 2

Supongamos ahora que se desea calcular el valor actuarial para un capital unitario pagadero al final de f/q , de un ente de edad (X) con tal de que suceda dentro de n años, para este caso el valor actual y actuarial es:

Pagadero la momento de f/q

En este caso el valor actual es

$$\mathbf{X}_{xn}^{-1} \begin{cases} v^t & \text{Para } t > 0 \text{ y } t < n \\ 0 & \text{Para el resto} \end{cases}$$

El valor actuarial y el riesgo de este modelo vienen dado por

$$E(\bar{\mathbf{x}}_{x:n}^{-1}) = \bar{A}_{x:n}^{-1} = \int_0^{\infty} v^t {}_tP_x u_{x+t} dt$$

$$Var(\bar{\mathbf{x}}_{x:n}^{-1}) = {}^2\bar{A}_{x:n}^{-1} - \bar{A}_{x:n}^{-2} \quad (3.67)$$

Pagadero al final de año f/q

El valor actual es

$$\bar{\mathbf{x}}_{x:n}^{-1} \begin{cases} v^{t+1} & \text{con probabilidad } {}_t/q_x \quad t = 0, 2, 3, \dots, n-1 \\ 0 & \text{con probabilidad } {}_n p_x \end{cases}$$

Puesto que t varia desde cero hasta n se puede utilizar la expresión 3.34 para calcular el valor actuarial de este modelo:

$$A_{x:n}^{-1} = E(\bar{\mathbf{x}}_{x:n}^{-1}) = \sum_{t=0}^{n-1} v^{t+1} {}_t/q_x = \sum_{t=0}^{n-1} v^{t+1} A_x = \frac{\sum_{t=0}^{n-1} C_{x+t}}{D_x} = \frac{\sum_{z=x}^{w+n-1} C_z}{D_x} = \frac{M_x - M_{x+n}}{D_x} \quad (3.68)$$

Nótese que $A^1_{x:n}$ es una generalización de ${}_t/1A^1$ para t variando desde cero hasta n. El riesgo que corre la aseguradora para este modelo viene dado por

$$\text{Var}(\mathbf{x}^1_{x:n}) = E(\mathbf{x}^1_{x:n})^2 - (A^1_{x:n})^2 = {}^2A^1_{x:n} - (A^1_{x:n})^2 \quad (3.69)$$

Modelo 3

Seguro de un capital unitario pagadero si este ocurre en cualquier momento a partir de la edad (X), a este seguro se lo denomina de vida entera.

Pagaderos al momento de f/g

Para este caso el valor actual es

$$\bar{\mathbf{x}}_x = v^t \quad \text{Para todo } t > 0$$

El valor actuarial y riesgo del modelo continuo son:

$$E(\bar{\mathbf{x}}_x) = \bar{A}_x = \int_0^{\infty} v^t {}_tP_x u_{x+t} dt \quad (3.70)$$

$$\text{Var}(\bar{\mathbf{x}}_x) = {}^2\bar{A}_x - \bar{A}_x^2 = \int_0^{\infty} v^{2t} {}_tP_x u_{x+t} - \left(\int_0^{\infty} v^t {}_tP_x u_{x+t} dt \right)^2$$

Pagadero a final del año f/q

El valor actual para el modelo discreto es

$$\bar{\mathbf{x}}_x = v^{t+1} \text{ con probabilidad } {}_t/ q_x \text{ para } t = 0, 1, 2, \dots, w-x$$

El valor actuarial y riesgo son respectivamente

$$E(\mathbf{x}_x) = A_x = \sum_{t=0}^{w-x} v^{t+1} {}_t/ q_x = \sum_{t=0}^{w-x} {}_t/1 A_x = \frac{M_x}{D_x}$$

$$Var(\mathbf{x}_x) = {}^2A_x - A_x^2 = \sum_{t=0}^{w-x} e^{-2\delta(t+1)} {}_t/ q_x - A_x^2 \quad (3.71)$$

Modelo 4

Seguro de un capital unitario pagadero si (x) supervive n años más, a este modelo se denomina capital diferido para caso de supervivencia.

El Valor Actual es:

$${}_n \mathbf{X}_x \begin{cases} 0 & \text{con probabilidad } {}_n q_x \\ v^n & \text{con probabilidad } {}_n p_x \end{cases}$$

Para el caso donde la indemnización se paga al final del año de f/q o al momento de f/q se pueden calcular de la siguiente forma el Valor actuarial y riesgo.

$$E({}_n \mathbf{x}_x) = A_{x:n}^{-1} = v^n {}_n P_x = v^n \frac{l_{x+n}}{l_x} = \frac{D_{x+n}}{D_x}$$

$$Var({}_n \mathbf{x}_x) = {}^2A_{x:n}^{-1} - A_{x:n}^{-2} \tag{3.72}$$

En este caso no existe distinción entre modelos continuos y discretos puesto que solo existe un solo pago, si el ente sobrevive n años.

3.3.3.2. Modelos con años de diferimiento

Modelo 1

Seguro de un capital unitario pagadero siempre que este ocurra transcurrido m años y dentro de los n años siguientes.

Pagadero al momento de f/g

El valor actual para este caso es

$${}_{m/n}\bar{\mathbf{X}}_x \begin{cases} 0 & t > 0 \text{ y } t < m \\ v^t & t > m \text{ y } t < m+n \\ 0 & t > m+n \end{cases}$$

El valor actuarial y riesgo son respectivamente

$${}_{m/n}\bar{A}_x = E({}_{m/n}\bar{\mathbf{X}}_x) = \int_m^{m+n} v^t {}_tP_x u_{x+t} dt$$

$$Var({}_{m/n}\bar{\mathbf{X}}_x) = {}_{m/n}\bar{A}_x^2 - {}_{m/n}A_x^2 \tag{3.73}$$

Pagadero al final del año de f/g

El valor Actual para el caso discreto es:

$${}_{m/n}\mathbf{X}_x \begin{cases} 0 & \text{con probabilidad } {}_m q_x \\ v^t & \text{con probabilidad } {}_t q_x \quad t = m, m+1 \dots m+n-1 \\ 0 & \text{con probabilidad } {}_{n+m} P_x \end{cases}$$

El valor actuarial es

$${}_{m/n}A_x = E({}_{m/n}\mathbf{x}_x) = \sum_{t=0}^{m+n-1} v^{t+1} {}_tq_x = \sum_{t=0}^{m+n-1} v^{t/1} A_x = \frac{M_{x+m} - M_{x+m+n}}{D_x} \quad (3.74)$$

Modelo 2

Seguro de un capital unitario pagadero siempre que el f/q ocurra una vez transcurrido m años

Pagadero al momento de f/q

El valor actual para el caso continuo es

$${}_{m/}\bar{\mathbf{x}}_x \begin{cases} 0 & t > 0 \text{ y } t < m \\ v^t & t > m \end{cases}$$

El valor actuarial y riesgo son respectivamente

$${}_{m/}\bar{A}_x = E({}_{m/}\bar{\mathbf{x}}_x) = \int_m^{\infty} v^t {}_tP_x u_{x+t} dt \quad (3.75)$$

$$Var({}_{m/}\bar{\mathbf{x}}_x) = {}_{m/}\bar{A}_x^2 - {}_{m/}A_x^2$$

Pagadero al final del año de f/q

El valor actual para el caso discreto es:

$${}_m/\mathbf{x}_x \begin{cases} 0 & \text{con probabilidad } {}_m q_x \\ v^{t+1} & \text{con probabilidad } {}_t q_x \end{cases} \quad t = m, m+1\dots$$

El valor actuarial es:

$${}_m/A_x = E({}_m/\mathbf{x}_x) = \sum_{t=0}^{\infty} v^{t+1} {}_t q_x = \sum_{t=0}^{\infty} {}_t/1 A_x = \frac{M_{x+m}}{D_x} \quad (3.76)$$

3.3.3.3. Modelos con indemnización incremental y sin años de diferimiento

Estos modelos tienen la característica de que a medida que el tiempo avanza y el suceso de f/q no acaece entonces la indemnización se incrementa en una unidad monetaria.

Modelo 1

Seguro de vida temporal creciente cuya prestación consiste en el pago de la prestación siempre que (x) acaezca dentro de los n años.

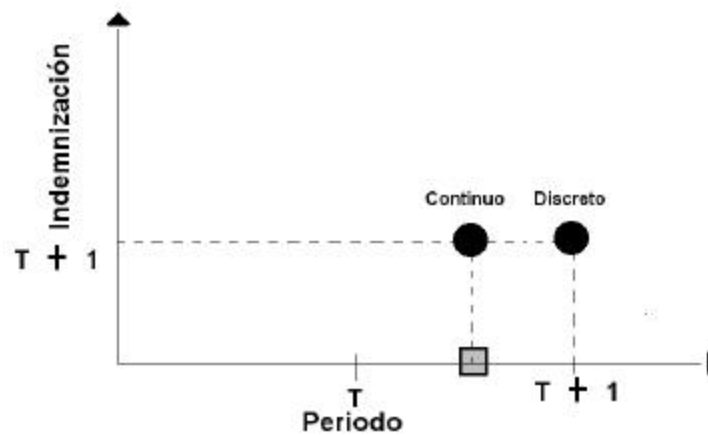


Figura 3.6. Diferencia en el pago entre seguros continuos y discretos con indemnización incremental.

Pagadero al momento de f/g

El valor actual para el caso continuo

$$(\bar{x}A)_{x:n}^1 \begin{cases} [t + 1] v^t & \text{para } t > 0 \text{ y } t < n \\ 0 & \text{en otro caso} \end{cases}$$

donde $[t + 1]$ representa el entero mayor no menor a $t + 1$

El valor actual y riesgo para este caso

$$\begin{aligned} (IA)_{x:n}^1 &= E\left[(\bar{\mathbf{x}}A)_{x:n}^1\right] = \int_0^n [t + 1]v^t {}_tP_x u_{x+t} dt \\ \text{Var}\left[(\bar{\mathbf{x}}A)_x\right] &= {}^2(I\bar{A})_{x:n}^1 - (IA)_{x:n}^1{}^2 \end{aligned} \quad (3.77)$$

Pagadero al final del año de f/g

Valor Actual

$$(\mathbf{x}A)_{x:n}^1 \begin{cases} v^{t+1} (t+1) & \text{con probabilidad } {}_tq_x \quad t = 0, 1, 2, \dots, n-1 \\ 0 & \text{con probabilidad } {}_mP_x \end{cases}$$

El valor actuarial es:

$$\begin{aligned} (IA)_{x:n}^1 &= E\left[(\mathbf{x}A)_{x:n}^1\right] = \sum_{t=0}^{n-1} (t+1)v^{t+1} {}_tq_x = \sum_{t=0}^{n-1} (t+1) {}_{t/1}A_x \\ &= \frac{R_x - R_{x+n} - nM_{x+n}}{D_x} \end{aligned} \quad (3.78)$$

La figura 3.6 nos muestra la diferencia existente entre el modelo discreto y continuo para este caso, mientras el discreto permite pagar el monto pactado al final del año de f/q , el modelo continuo permite pagar el mismo monto que el discreto pero no al final del año si no en el momento mismo del fallecimiento, este efecto lo causa la parte entera del tiempo futuro de supervivencia (entero mayor) que se definió en el modelo continuo.

Modelo 2

Modelo de seguro de vida entera creciente, que proporciona una anualidad de 1 u.m si el fallecimiento o quiebra ocurre dentro del primer años, 2 u.m si ocurre entre el primero y segundo año, y así sucesivamente, creciendo en 1 u.m hasta el final hasta que el ente f/q .

Pagadero al momento de f/q

El valor actual para este modelo es

$$\left(\overline{\mathbf{x}A}\right)_x = [t + 1] v^t \quad \text{para todo } t > 0$$

El valor actuarial y riesgo es:

$$(I\bar{A})_x = E[(\mathbf{x}\bar{A})_x] = \int_0^{\infty} (t+1)v^t {}_tP_x u_{x+t} dt \quad (3.79)$$

$$Var[(\mathbf{x}\bar{A})_x] = {}^2(I\bar{A}_x) - (I\bar{A}_x)^2$$

Nótese que la primera impresión que nos da este modelo es que el asegurado va a recibir una cantidad colosal o infinita de dinero en un tiempo dado, si es que el ente f/g después de un largo periodo de tiempo, pero realmente no es así puesto que la vida de un ente nunca es infinita, siempre existe una cota superior w por el cual más allá de esa edad el ente no puede sobrevivir, es decir la

$$P(T > w) = 0$$

Pagadero al final del año de f/g

El Valor Actual es:

$$(\mathbf{x}A)_x = [t+1] v^t \quad \text{con probabilidad} \quad {}_t/q_x \quad t = 0,1,2,\dots,w-x$$

El valor actuarial sería

$$(IA)_x = E[(\mathbf{x}A)_x] = \sum_{t=0}^{w-x} (t+1)v^t {}_t/q_x = \sum_{t=0}^{w-x} {}_t/q_x A_x = \frac{R_x}{D_x} \quad (3.80)$$

3.3.3.4. Modelos con indemnización incremental y con años de diferimiento

Modelo 1

Modelo de seguro Incremental pagadero si el individuo f/q después de m años.

Pagadero al momento de f/q

El Valor Actuarial y riesgo son respectivamente:

$$\begin{aligned} {}_{m/}(\bar{IA})_x &= E[{}_{m/}(\bar{IA})_x] = \int_m^{\infty} (t - m + 1)v^t {}_tP_x u_{x+t} dt \\ \text{Var}[{}_{m/}(\bar{IA})_x] &= {}_{m/}^2(\bar{IA})_x - ({}_{m/}(\bar{IA})_x)^2 \end{aligned} \quad (3.81)$$

Pagaderos al final del año de f/q

El Valor actuarial es

$${}_{m/}(IA)_x = \sum_{t=m}^{w-x-m} (t+1-m)v^{t+1} {}_tq_x = \sum_{t=m}^{w-x-m} (t+1-m) {}_tA_x = \frac{R_{x+m}}{D_x} \quad (3.82)$$

Modelo 2

Modelo de seguro Incremental pagadero si el ente f/q después de m años hasta los n años siguientes.

Pagadero al momento de f/q

Valor actuarial y riesgo

$$\begin{aligned} {}_{m/n}(\bar{IA})_x &= E[{}_{m/n}(\bar{IA})_x] = \int_m^{m+n} (t-m+1)v^t {}_tP_x u_{x+t} dt \\ \text{Var}[{}_{m/n}(\bar{IA})_x] &= {}_{m/n}^2(\bar{IA})_x - ({}_{m/n}(\bar{IA})_x)^2 \end{aligned} \quad (3.83)$$

Pagadero al final del año de f/q

El valor actuarial para este caso es

$$\begin{aligned} {}_{m/n}(IA)_x &= \sum_{t=m}^{m+n-1} (t+1-m)v^{t+1} {}_t/q_x = \sum_{t=m}^{m+n-1} (t+1-m) {}_t/A_x \\ &= \frac{R_{x+m} - R_{x+m+n} - nM_{x+m+n}}{D_x} \end{aligned} \quad (3.84)$$

3.3.3.5. Otros modelos de seguros

Modelo 1

Modelo de seguros de vida entera que proporciona prestaciones crecientes al final del m – ésimo año en que acaezca f/q , es decir $1/m$ u.m por f/q al final del primer m – ésimo año, $2/m$ u.m si el f/q acaece en el segundo m – ésimo año, etc.

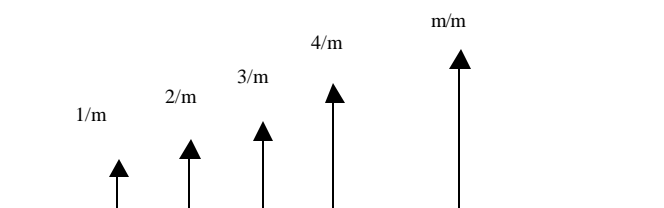


Figura 3.7 Indemnización de un seguro fraccionario dentro del primer año

Las indemnizaciones presentadas en la figura 3.7 corresponden al inicio del año, vemos que se encuentran particionado en m partes, procedemos a obtener el valor medio de las indemnizaciones de la siguiente manera:

$$ValMedio = \sum_{k=0}^m \frac{k}{m} = \frac{1}{m} \sum_{k=1}^m k = \frac{(m+1)}{2m} \quad (3.85)$$

Tomando el valor medio para cada año tenemos que el valor actuarial es:

$$\begin{aligned} (I^m A)_x &\approx \sum_{t=0}^{\infty} \left(t + \frac{m+1}{2m} \right) v^{t+1} {}_t|q_x = \sum_{t=0}^{\infty} \left(t + \frac{m+1}{2m} \right) {}_t|1 A_x \\ &= (IA)_x - \frac{m-1}{2m} A_x \end{aligned} \quad (3.86)$$

Pagadero al momento de f/q

La prima única neta o valor actuarial en este caso es

$$(I^m \bar{A})_x \approx (I\bar{A})_x - \frac{m-1}{2m} \bar{A}_x \quad (3.87)$$

Si $m \rightarrow \infty$ tenemos que

$$(\bar{I}\bar{A})_x = \int_0^{\infty} t v^t {}_t P_z u_{x+t} dt = (I\bar{A})_x - \frac{1}{2} \bar{A}_x \quad (3.88)$$

Generalizando, si tenemos una función de indemnización continua $g(t)$ pagadero al momento de f/q tenemos que su valor actuarial es:

$$(g\bar{A})_x = \int_0^{\infty} g(t) v^t {}_t P_z u_{x+t} dt \quad (3.89)$$

Notemos que para la expresión 3.88 $g(t)=t$

Modelo 2

Modelo de seguros decreciente temporal que proporciona una prestación inicial de n unidades monetarias decreciendo cada año 1 u.m. con ningún pago si el f/q acaece luego de los n años.

Pagadero al momento de f/q

El valor actuarial y riesgo son

$$\begin{aligned} (D\bar{A})_{x:n}^1 &= \int_0^n (n-t)v^t {}_tP_x u_{x+t} dt \\ \text{Var}[(D\bar{A})_{x:n}^1] &= (D\bar{A})_{x:n}^1 - (D\bar{A})_{x:n}^1{}^2 \end{aligned} \quad (3.90)$$

Pagadero al final del año de f/q

El valor actuarial para este caso es

$$\begin{aligned} (DA)_{x:n}^1 &= \sum_{t=0}^{n-1} (n-t)v^{t+1} {}_{t/q_x} = \sum_{t=0}^{n-1} (n-t) {}_{t/1}A_x \\ &= \frac{nM_x - (R_{x+1} - nR_{x+n+1})}{D_x} \end{aligned} \quad (3.91)$$

Se pueden desarrollar modelos combinando dos o más de los que se han expuesto hasta aquí, por ejemplo:

- ✓ Modelo de seguro de una unidad monetaria pagadero al momento o al final del año de f/q si es que el ente (X) acaece dentro de los n primeros años, y si este sobrevive los n años. (Modelo 2 y 4).

- ✓ Modelo de seguros de un capital unitario si este acaece dentro de los m primeros años, y a partir de m se paga en forma creciente comenzando con una unidad monetaria en cualquier momento en que acaezca el ente. (Modelo 2, Modelo 8)

- ✓ Modelo de seguros de vida entera donde el valor de la indemnización esta regido por la función continua $g(t) = \text{Sen}(t)$ pagadero al momento de f/q .

El cálculo del valor actuarial del último ejemplo es factible si se conoce la ley analítica que rige la tasa de mortalidad, así como los modelos donde el pago se lo hace al momento de f/q , pero como se mencionó en la realidad se trabaja frecuentemente con tablas de mortalidad, por lo que se debe buscar una forma de aproximar

los valores actuariales de los modelos continuos utilizando las tablas de mortalidad.

3.3.4. Relación entre los modelos pagaderos al momento de f/q y los pagaderos al final del año de f/q

Se va a particularizar la demostración con el modelo de seguro de vida entera de la siguiente forma

Sabemos que

$$A_x = \int_0^{\infty} v^t {}_tP_z u_{x+t} dt \quad (3.92)$$

Particionando la región de integración en infinitas partes e integrando sobre ella y luego sumándolas tenemos que la expresión anterior también es igual a:

$$A_x = \int_0^{\infty} v^t {}_tP_z u_{x+t} dt = \sum_{k=0}^{\infty} \int_k^{k+1} v^t {}_tP_x u_{x+t} dt \quad (3.93)$$

Como se va a suponer distribución uniforme de las muertes nos conviene realizar el siguiente cambio de variable $s = t - k$ con esto tenemos el siguiente resultado.

$$\begin{aligned}
\bar{A}_x &= \sum_{k=0}^{\infty} \int_k^{k+1} v^t {}_tP_x u_{x+t} dt = \sum_{K=0}^{\infty} \int_0^1 v^{k+s} {}_{k+s}P_x u_{x+k+s} ds \\
&= \sum_{k=0}^{\infty} v^{k+1} {}_sP_x \int_0^1 v^{s-1} {}_sP_{x+k} u_{x+k+s} ds \quad (3.94)
\end{aligned}$$

Bajo la suposición de distribución uniforme de muertes sobre las edades tenemos que ${}_sP_{x+k} u_{x+k+s} = q_{x+k}$ entonces.

$$\bar{A}_x = \sum_{k=0}^{\infty} v^{k+1} {}_kP_x q_{x+k} \int_0^1 (1+i)^{1-s} ds = \frac{i}{d} A_x \quad (3.95)$$

Así generalizando, si \bar{A} es el valor actuarial de un modelo de seguro pagadero al momento de f/q de un ente de edad (X) y A es el valor actuarial de un modelo de seguro para el mismo ente pagadero al final del año de f/q entonces

$$\bar{A} \approx \frac{i}{d} A \quad (3.96)$$

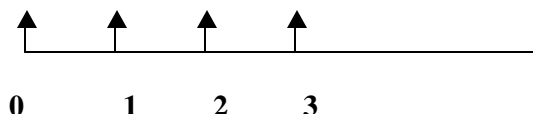
3.4. Valores Actuariales de las prestaciones para los casos de supervivencia.

En la sección anterior se considero modelos actuariales, donde se indemnizaba a un ente (x) cuando acaecía el f/q de este, en esta sección se desarrollarán modelos para prestaciones donde se consideran las supervivencias de los entes.

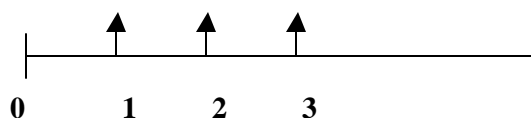
Una anualidad de vida es una serie de pagos realizados continuamente o en intervalos iguales de tiempo (quincenal, mensual, trimestral, anual etc) mientras el individuo sobreviva, estos pagos pueden ser temporales, o pagaderos por toda la vida.

Los pagos pueden ser de dos tipos, anticipado o vencido, una anualidad es anticipada cuando los pagos se lo realizan al inicio de cada intervalo de tiempo, mientras que cuando se realiza en forma vencida se hace el pago al final del intervalo como se muestra en la siguiente figura:

Pago Anticipado



Pago Vencido



Cuando la longitud de los intervalos son extremadamente pequeños, esto es los pagos se los realiza de forma continua, no existe diferencia alguna si los pagos se lo realiza en forma anticipada o vencida.

3.4.1. Valor actuarial

Supongamos que se desea obtener el valor actual de pagos continuos de una unidad monetaria de un individuo (x) en un periodo de capitalización de n años.

$$\bar{a}_n = \int_0^n v^s ds$$

(3.97)

Se sabe que el tiempo futuro de supervivencia es la variable aleatoria T , entonces si el periodo de capitalización viene dado por esta variable tenemos que el valor actual es.

$$\bar{a}_{\bar{T}} = \int_0^T v^s ds = \frac{1-v^T}{d} \quad (3.98)$$

Este es el valor actual de los modelos de anualidades de vida y como se definió antes el valor actuarial es igual al valor esperado del valor actual, así se tiene que:

$$E(\bar{a}_{\bar{T}}) = E\left(\frac{1-v^T}{d}\right) = \frac{1}{d}(1-E(v^T)) \quad (3.99)$$

Se puede deducir que los valores actuariales de las anualidades de vida también se los puede expresar en termino de los valores actuariales de los modelos de seguros, como más adelante se verificará.

El riesgo es otra característica importante de estos modelos y como se mostró, es la varianza del valor actual definido en 3.97 así tenemos que el riesgo es:

$$Var(\bar{a}_{\bar{T}}) = Var\left(\frac{1-v^T}{d}\right) = \frac{1}{d^2} Var(v^T) \quad (3.100)$$

Al igual que la esperanza matemática, el riesgo de los modelos de las rentas quedan en términos del factor de actualización que se presento en los modelos de seguros, como se aprecia en la fórmula 3.100 .

El método para el cálculo de la esperanza matemática para las anualidades discretas es muy similar al continuo.

3.4.2. Modelos de Anualidades o Rentas de Vida

3.4.2.1. Modelos de rentas sin años de diferimiento

Modelo 1

Renta vitalicia unitaria con pagos de por vida de forma inmediata y anticipada

Pagos continuos

El valor actual seria: $\bar{a}_{\bar{T}}$ Para todo $T > 0$

$$\text{El valor actuarial es: } \bar{a}_x = \int_0^{\infty} \bar{a}_{\bar{t}} P_x u_{x+t} dt \quad (3.101)$$

Integrando la ultima expresión tenemos lo siguiente

$$\bar{a}_x = \int_0^{\infty} \bar{a}_{\bar{t}} P_x u_{x+t} dt = \int_0^{\infty} v^t {}_t p_x dt \quad (3.102)$$

Esta última expresión es importante ya que facilita los cálculos e interpretación del valor actuarial, y se generaliza para los restantes modelos de pagos continuos, variando sólo los límites integración.

Utilizando el resultado obtenido en 3.98 se puede expresar el valor actuarial de la anualidad de vida entera en términos del valor actuarial del modelo de seguro de vida entera, como se muestra a continuación.

$$\bar{a}_x = E\left(\frac{1-v^T}{d}\right) = \frac{1}{d}(1 - E(v^T)) = \frac{1}{d}(1 - \bar{A}_x) \quad (3.103)$$

De aquí tenemos la siguiente igualdad.

$$d\bar{a}_x + A_x = 1 \quad (3.104)$$

Así mismo el riesgo de este modelo es igual a:

$$\text{Var}(\bar{a}_T) = \text{Var}\left(\frac{1-v^T}{d}\right) = \frac{1}{d^2} \text{Var}(v^T) = \frac{1}{d^2} ({}^2A_x - A_x^2) \quad (3.105)$$

Pagos Anuales

Definamos el valor actuarial de una renta anticipada como \ddot{a} y a la vencida sin las diéresis, así tenemos entonces que:

El valor actual es

$$\ddot{a} = a_{\bar{T}} \quad \text{con probabilidad } {}_{t-1}q_x \quad \text{para } t=1,2,3,4\dots w-x$$

Así tenemos que el valor actuarial en términos de los símbolos de conmutación es.

$$\ddot{a}_x = \sum_{t=1}^{w-x} a_{t-1|} q_x = \sum_{t=0}^{w-x} E_x = \frac{1}{D_x} \sum_{t=0}^{w-x} D_{x+t} = \frac{N_x}{D_x} \quad (3.106)$$

Para el caso vencido el análisis es similar pero desplazando los pagos 1 unidad hacia la derecha, entonces el valor actuarial sería

$$a_x = \frac{N_{x+1}}{D_x} \quad (3.107)$$

Modelo 2

Renta vitalicia anual, unitaria inmediata y temporal por n años.

Pagos continuos

Valor actual $\bar{a}_{\overline{T}}$ con probabilidad para $t > 0$ y $t < n$

El valor actuarial $\bar{a}_{x:n} = \int_0^n v^t {}_t p_x dt$

El riesgo para este modelo es igual

$$\text{Var}(\bar{a}_T) = \frac{1}{d^2} \left({}^2\bar{A}_{x:n} - \bar{A}_{x:n}^2 \right) \quad (3.108)$$

Pagos Anuales

El valor actual de la renta anticipada es:

$$\ddot{\mathbf{a}} \left\{ \begin{array}{l} \ddot{a}_{\bar{T}} \quad \text{Con Probabilidad} \quad {}_{t-1}q_x \quad t=1, 2, 3, \dots, n-1 \\ \ddot{a}_{\bar{n}} \quad \text{Con Probabilidad} \quad {}_{n-1}p_x \end{array} \right.$$

El valor actuarial vendría dado por la siguiente Expresión

$$\ddot{a}_x = \sum_{t=1}^{n-1} \ddot{a}_{t-1} q_x = \sum_{t=0}^{n-1} {}_tE_x = \frac{1}{D_x} \sum_{t=0}^{n-1} D_{x+t} = \frac{N_x - N_{x+n}}{D_x} \quad (3.109)$$

El valor actuarial de la renta temporal inmediata vencida es:

$$a_x = \sum_{t=1}^n a_{t-1} q_x = \sum_{t=0}^n {}_tE_x = \frac{1}{D_x} \sum_{t=0}^n D_{x+t} = \frac{N_{x+1} - N_{x+n+1}}{D_x} \quad (3.110)$$

3.4.2.2. Modelos de rentas con años de diferimiento

Modelo 1

Renta vitalicia, unitaria, diferida a n años, anticipada e ilimitada.

Pagos Continuos

El valor actual para este modelo es $\bar{a}_{\overline{t}|}$ Para todo $t > n$

El Valor Actuarial es:

$${}_{n|}\bar{a}_x = \int_n^{\infty} v^t p_x dt \quad (3.111)$$

El riesgo es

$$Var(\bar{a}_{\overline{t}|}) = \frac{1}{d^2} Var(v^T) = \frac{1}{d^2} \left({}^2_{n|}\bar{A}_x - n| \bar{A}^2 \right) \quad (3.112)$$

Pagos Anuales

El valor actual se define como

$${}_{n/}\ddot{\mathbf{a}} \begin{cases} 0 & \text{Con probabilidad } {}_n/ q_x \\ \ddot{\mathbf{a}}_{\overline{t-n}} & \text{Con probabilidad } {}_{t-1}/ q_x \quad t = n+1, n+2, \dots, w-x \end{cases}$$

El valor actuarial con los pagos Anticipados es:

$${}_{n/}\ddot{a}_x = \sum_{t=n+1}^{w-x} \ddot{\mathbf{a}}_{\overline{t-1}} q_x = \sum_{t=n}^{w-x-1} {}_t E_x = \frac{1}{D_x} \sum_{t=n}^{w-x-1} D_{x+t} = \frac{N_{x+n}}{D_x} \quad (3.113)$$

El valor actuarial con pagos vencidos es:

$${}_{n/}a_x = \sum_{t=n}^{w-x} \mathbf{a}_{\overline{t-1}} q_x = \sum_{t=n+1}^{w-x-1} {}_t E_x = \frac{1}{D_x} \sum_{t=n+1}^{w-x-1} D_{x+t} = \frac{N_{x+n+1}}{D_x} \quad (3.114)$$

Modelo 2

Renta vitalicia unitaria, diferida n años y temporal por m años

Pagos continuos

El Valor Actual es $\overline{a}_{\overline{t}}$ para $t > n$ y $t < m$

$$\text{El valor actuarial es: } {}_{n/m}\overline{a}_x = \int_n^{n+m} v^t {}_t p_x dt \quad (3.115)$$

El riesgo es:

$$Var(\bar{a}_T) = \frac{1}{d^2} Var(v^T) = \frac{1}{d^2} \left({}_{n/m}^2 \bar{A}_x - {}_{n/m} \bar{A}^2 \right) \quad (3.116)$$

Pagos Anuales

El valor actuarial con pagos anticipados es

$$\begin{aligned} {}_{n/m} \ddot{a}_x &= \sum_{t=n+1}^{n+m-1} a_{t-1|q_x} = \sum_{t=n}^{n+m-1} E_x = \frac{1}{D_x} \sum_{t=n}^{n+m-1} D_{x+t} \\ &= \frac{N_{x+n} - N_{x+n+m}}{D_x} \end{aligned} \quad (3.117)$$

Valor Actuarial con pagos vencido se define como

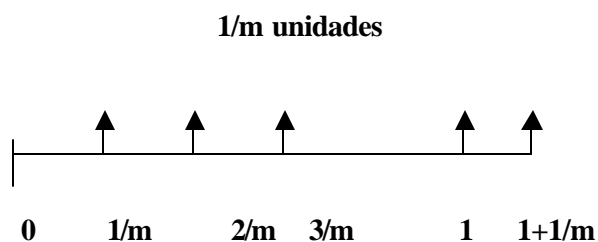
$$\begin{aligned} {}_{n/m} a_x &= \sum_{t=n}^{n+m-1} a_{t-1|q_x} = \sum_{t=n+1}^{n+m} E_x = \frac{1}{D_x} \sum_{t=n+1}^{n+m} D_{x+t} \\ &= \frac{N_{x+n} - N_{x+n+m}}{D_x} \end{aligned} \quad (3.118)$$

Comúnmente los pagos de las rentas se los realiza de manera mensual y anual pero no continua, para esto se propone el siguiente modelo.

3.4.2.3. Otros modelos de rentas

Modelo 3

Rentas vitalicias fraccionadas en pagadero cada m parte del año $1/m$ unidades monetarias, en forma inmediata.



El valor actuarial de este modelo sería

$$a_x^{(m)} = \frac{1}{m} \sum_{n=0}^{\infty} \sum_{t=1}^m {}_{n+\frac{t}{m}} E_x \quad (3.119)$$

Para aproximar ${}_{n+\frac{t}{m}} E_x$ interpolamos linealmente seleccionando dos puntos con $t = 0$ y $t = m$ entonces tenemos que:

$${}_{n+\frac{t}{m}} E_x \approx \frac{t}{m} ({}_{n+1} E_x - {}_n E_x) + {}_n E_x \quad (3.120)$$

Reemplazando esta expresión en 3.119 y realizando algunas Operaciones matemáticas tenemos

$$a_x^{(m)} = \frac{1}{m} \sum_{n=0}^{\infty} \sum_{t=1}^{\infty} \left[\frac{t}{m} ({}_{n+1}E_x - {}_nE_x) + {}_nE_x \right] = \ddot{a}_x - \frac{m+1}{2m} \quad (3.121)$$

Observe que la expresión 3.121 queda en términos de una renta anual con pagos anticipados, pero se puede expresar también en términos de una renta anual pagadero en forma vencida como.

$$a_x^{(m)} = a_x + 1 - \frac{m+1}{2m} = a_x - \frac{m-1}{2m} \quad (3.122)$$

Así tenemos que los restantes modelos de rentas fraccionadas en termino de las rentas anuales son:

$$\begin{aligned} {}_n/a_x^{(m)} &\approx {}_n/a_x \frac{m+1}{2m} {}_nE_x \\ a_{x:n}^{(m)} &\approx a_x^{(m)} - {}_n/a_x^{(m)} \approx a_{x:n} + \frac{m-1}{2m} (1 - {}_nE_x) \end{aligned} \quad (3.123)$$

3.4.3. Capitalización Actuarial

Si cada elemento del colectivo / sector realiza el pago de 1 u.m. a un determinado fondo al final de cada año si sobrevive y estos pagos se capitalizan en el fondo hasta el final de n años, la parte final de cada superviviente al final de los n años se simboliza $S_{x:n}$ por lo tanto para el valor actuarial del montante al final de los n años de supervivencia, con pago unitario, pos - pagable, se tendrá en cuenta lo siguiente:

El primer pago hecho en $(x+1)$ se convierte al final en $(x+n)$

$$\frac{1}{{}_{n-1}E_{x+1}}$$

El segundo pago hecho en $(x+2)$ se convierte al final en $(x+n)$

$$\frac{1}{{}_{n-2}E_{x+2}}$$

Por lo tanto el montante actuarial para cada superviviente a la edad $(x+n)$ vendría dado por.

$$\begin{aligned}
 S_{x:n} &= \sum_{t=1}^n \frac{1}{n-t} E_{x+t} = \frac{N_{x+1} - N_{x+n+1}}{D_{x+n}} \\
 S_{x:n}^{\bullet\bullet} &= \frac{N_x - N_{x+n}}{D_{x+n}} \quad (3.124)
 \end{aligned}$$

3.5 Valores Actuariales de las Primas Netas

En esta sección vamos a juntar las metodologías para calcular los valores actuariales de los seguros de vida y anualidades de vida para trabajar conjuntamente con ellos, la idea es que la aseguradora emite una póliza de seguros y el asegurado tiene que realizar pagos o primas en intervalos de tiempo iguales por la adquisición de esta póliza, los pagos se convierten en anualidades de vida o rentas vitalicias, ya que el asegurado cancela la prima mientras vive.

Para establecer el pago que tiene que hacer el asegurado para tener acceso a los beneficios que estipula el contrato se debe considerar el “**Principio de Equilibrio Financiero Actuarial**” el cual afirma que la prima del asegurado debe ser tal que la pérdida esperada de la aseguradora debe ser 0, esto es $E(\mathbf{I}) = 0$ donde \mathbf{I} es la variable que denota la pérdida.

Así si p es el pago que hace el asegurado (prima única pura) y X el valor actual de la indemnización que cancela la aseguradora si el ente f/q entonces la pérdida de la aseguradora sería $I = X - p$ de aquí tenemos que:

$$\begin{aligned} E(I) &= E(X - p) = E(X) - p = 0 \\ E(X) &= p \end{aligned} \quad (3.125)$$

Cuando el pago no se lo realiza una sola vez, si no en forma periódica (mensual, quincenal, anual, etc.) estos pagos se los denomina primas netas y se los representa con P , además si los pagos de estas prima se los realiza en intervalos menores a un año se dice que las primas son "Primas netas fraccionadas".

Los pagos de las primas son en general anticipadas puesto que el primer pago se cancela en el momento que se firma el contrato y los otros en pagos periódicos iguales mientras hay supervivencia o hasta que se cumpla alguna característica del contrato, también se habla de "primas recargadas o brutas" cuando éstas incluyen todos los gastos inherentes a la emisión y montaje de la póliza.

Tenemos que la variable aleatoria I (También llamada función de pérdida) para el caso discreto y continuo son respectivamente.

$$\begin{aligned} \bar{I}(t) &= v^t - \bar{P}a_{\bar{t}} && \text{Caso continuo} \\ I(t) &= v^{t+1} - Pa_{\overline{t+1}} && \text{Caso discreto} \end{aligned} \quad (3.126)$$

En forma general, la prima neta es igual al valor esperado de 3.126

$$\begin{aligned} E[I(t)] &= E(v^t - Pa_{\overline{t+1}}) = 0 \equiv E(v^t) - PE(a_{\bar{t}}) = 0 \\ \Rightarrow P &= \frac{E(v^t)}{E\left(\frac{1-v^t}{d}\right)} \end{aligned} \quad (3.127)$$

Y la varianza de la función de pérdida es:

$$\begin{aligned} \text{Var}[I(t)] &= \text{Var}(v^t - Pa_{\overline{t+1}}) = 0 \equiv \text{Var}\left(\frac{v^t(P+d) - P}{d}\right) \\ &= \left(1 + \frac{P}{d}\right)^2 \text{Var}(v^t) \end{aligned} \quad (3.128)$$

Una vez expuesto en forma general lo que son las primas netas, vamos a considerar algunas operaciones particulares de seguros y anualidades para así calcular las obligaciones por periodo del asegurado.

3.5.1. Valor actuarial de la prima neta, para operaciones de seguros y rentas vitalicias

Para un seguro de vida completa con pagos de primas anticipado de manera inmediata.

Principio de equilibrio $P_x \ddot{a}_x = A_x$ (3.129)

Notemos que en la primera parte de la igualdad tenemos las obligaciones del asegurado que no es más que el valor actuarial para el modelo de anualidad de vida entera ya que el asegurado paga la prima mientras viva, y en la parte izquierda las obligaciones de la aseguradora, la prima neta para este caso es.

$$P_x = \frac{A_x}{\ddot{a}_x} = \frac{M_x}{N_x} \quad (3.130)$$

Supongamos ahora que en el contrato se estipula que el asegurado debe de cancelar sus obligaciones por el seguro de vida entera hasta un tiempo t determinado, entonces así tenemos que la prima a pagar es:

$$P_x = \frac{A_x}{\ddot{a}_{x:t}} \quad (3.131)$$

Siguiendo el mismo análisis tenemos entonces la siguiente tabla de primas:

Tabla I
Simbología actuarial para los planes

Planes para una persona de edad (x)	Formula de prima
1. Seguro temporal por n años, con primas pagaderas hasta	$P_{xn}^1 = \frac{A_{x:n}^1}{a_{x:n}}$
2. Seguro diferido m años y primas pagaderas hasta t años t < m.	${}_{m/}P_x = \frac{{}_{m/}A_x}{a_{xt}}$
3. Seguro incremental de vida entera y prima pagadera hasta t años.	$IP_x = \frac{(IA)_x}{a_{xt}}$
4. Seguro temporal por n años, e incremental a partir de los n años hasta los n+m años, con primas pagaderas hasta los n años	$P_x = \frac{A_{xn} + {}_{n/m}(IA)_x}{a_{x:n}}$

Como se deduce el número de planes existente solo depende de la creatividad que se tenga para combinar modelos de rentas y seguros. De la misma manera para un modelo de rentas tenemos.

Valor Actuarial de la prima neta, para una Renta vitalicia diferida por m años, donde las primas netas son pagadas hasta un tiempo $t < n$ tenemos:

Principio de equilibrio

$${}_tP_{(m/a_x)} \ddot{a}_{x:\overline{t}|} = m \ddot{a}_{x:\overline{t}|} \quad (3.132)$$

Entonces la prima neta para este caso seria:

$${}_tP_{(m/a_x)} = \frac{m \ddot{a}_{x:\overline{t}|}}{\ddot{a}_{x:\overline{t}|}} \quad (3.133)$$

Este plan requiere que los asegurados aporten una prima durante t años, y a partir de esto el asegurado recibirá una anualidad de 1 u.m mientras viva, si $t = m$ las características de este modelo son muy parecidas a los fondos de pensiones, ya que los trabajadores realizan aportaciones a un fondo, así cuando llegue el periodo de retiro laboral este recibirá una renta de forma anual o mensual para su sustento, claro está que un fondo considera más eventos que el fallecimiento del individuo, como es el retiro por invalidez, por despido etc, pero este modelo es un buen punto de partida.

3.5.2. Valor actuarial, para operaciones de seguros y rentas vitalicias que incluyen la devolución o retención de primas

Algunos planes incluyen en la póliza la devolución o la retención de las primas no pagadas si es que el asegurado f/q antes del tiempo contemplado para cancelar las primas, para ilustrar esto tenemos los siguientes ejemplos.

Una persona (40) años contrata un seguro de vida entera de 50.000 pagadero al final del año de f/q mediante el pago de primas anuales durante los 10 primeros años.

- a) con la estipulación de que la indemnización es pagada en cualquier año de fallecimiento.

- b) Si el fallecimiento ocurre antes de los 49 años se restara de la indemnización pactada el valor de las primas no pagadas.

Caso a

Este caso es el que ya se ha considerado en (3.130), la prima neta sería.

$${}_9P_{40} = 50000 \left(\frac{A_{40}}{\ddot{a}_{40:9}} \right)$$

Caso b

Este caso difiere de los planteados, puesto que se requiere encontrar el valor actuarial de las primas no pagadas, entonces tenemos que el principio de equilibrio actuarial será:

A_{40} - [Valor actuarial de primas no pagadas] \longrightarrow Obligación de la aseguradora

$P \ddot{a}_{40:9}$ \longrightarrow Obligación del asegurado

Tenemos entonces que el valor actuarial de las primas no pagadas es:

$$P \sum_{K=0}^9 v^t - P \ddot{a}_{40:9} = P \left[\left(\frac{1-v^{10}}{1-v} \right) - \ddot{a}_{40:9} \right]$$

Remplazando en el principio de equilibrio actuarial se tiene

$$P \ddot{a}_{\overline{40|9}} = A_{40} - P \left[\left(\frac{1-v^{10}}{1-v} \right) \ddot{a}_{\overline{40|9}} \right]$$

De aquí tenemos que la prima neta es:

$$P = 50000 A_{40} \left(\frac{1-v}{1-v^{10}} \right)$$

¿Cuanto tiene que cancelar anualmente un empleado de 40 años por tener acceso a una renta de jubilación vitalicia a partir de 65 años cuyo monto sea 10 veces su salario promedio con primas pagaderas hasta el año de jubilación?.

El principio de equilibrio actuarial sería

$$P \ddot{a}_{\overline{40|25}} = 10 \overline{S}_{25|} \ddot{a}_{40}$$

Y la prima neta derivada de esta expresión es

$$P \ddot{a}_{\overline{40|25}} = \frac{10 \overline{S}_{25|} \ddot{a}_{40}}{\ddot{a}_{\overline{40|25}}}$$

Calcular la prima neta del ejemplo anterior si además la cobertura incluye un sólo pago de 2000 al final del fallecimiento si este ocurre antes de los 65 años más 4 pagos anuales a sus deudos de 5000.

El principio de equilibrio financiero actuarial sería

$$P \ddot{a}_{\overline{40:25}} = 10 \overline{S}_{25/} \ddot{a}_{40} + 2000 A_{\overline{40:25}}^1 - [\text{Valor actuarial de los 4 pagos}]$$

El valor actual se lo define como

$$x = 5000 \sum_{t=0}^3 v^{t+1} \quad \text{Con probabilidad } {}_t/ q_{40} \quad T = 0,1,2,3$$

El valor actuarial sería

$$\begin{aligned} E(x) &= 5000 \sum_{t=0}^{24} v^{t+1} (1 + v + v^2 + v^3) {}_t/ q_{40} = 5000 \frac{1 - v^4}{1 - v} \sum_{t=0}^{24} {}_t/ q_{40} \\ &= 5000 \frac{1 - v^4}{1 - v} A_{\overline{40:25}} \end{aligned}$$

Reemplazando en el equilibrio financiero actuarial y despejando la prima neta tenemos que

$$P \ddot{a}_{40:25} = 10\bar{S}_{25/} \ddot{a}_{40} + 2000A_{40:25} + 5000 \frac{1-v^4}{1-v} A_{40:25}$$

$$P = \frac{10\bar{S}_{25/} \ddot{a}_{40} + 2000A_{40:25} + 5000 \left(\frac{1-v^4}{1-v} \right) A_{40:25}}{\ddot{a}_{40:25}}$$

3.5.3. Primas netas fraccionadas

Cuando el asegurado para tener acceso a una cobertura de seguro o de rentas de supervivencia, hace pagos con una frecuencia de m veces al año, la prima se denomina prima fraccionada. Existen dos clases de primas fraccionadas.

Con carácter liberatorio.- En este caso el asegurado no hace ningún descuento a las indemnización pactada por las primas fraccionadas no pagadas durante el resto del año de fallecimiento, se lo representa por $P^{(m)}$.

Supongamos que se desea calcular lo que debe cancelar un individuo (x) mensualmente por un seguro de vida entera.

$$P^{(m)} a_x^{(m)} = A_x \Rightarrow P^{(m)} = \frac{A_x}{a_x^{(m)}} \quad (3.134)$$

Los pagos en las m fracciones del año vendrían dado por $P^{(m)}/m$

Sin carácter liberatorio.- En este caso el asegurado descuenta de la indemnización pactada el monto de las primas fraccionadas no pagadas durante el resto del año de fallecimiento se representa por $P^{[m]}$

Por ejemplo para un seguro de vida entera, cuya prima se paga mensualmente tenemos que el principio de equilibrio financiero actuarial es:

$$P_x^{[m]} \ddot{a}_x = A_x - \frac{m-1}{2m} A_x P^{[m]} \quad (3.135)$$

De aquí tenemos que la prima neta sin carácter liberatorio es

$$P_x^{[m]} = \frac{A_x}{\ddot{a}_x^{[m]} + A_x \left(\frac{m-1}{2m} \right)} \quad (3.136)$$

3.5.4. Tipos de primas

Primas Recargadas.- Están compuestas por la prima pura más un recargo de seguridad, el cual recoge la varianza de la siniestralidad esperada.

Primas de Inventario.- Están compuestas por las primas puras y los gastos de gestión interna (sueldos, salarios, etc.)

Primas Tarifa Comercial.- Están compuestas por las primas puras más el recargo de gestión interna, más los gastos de gestión externa (gastos de montaje, gastos de gestión de cartera, recibos, etc) más los gastos de seguridad y recargos de beneficios.

Primas de Zillmer.- Están compuestos por la prima de recargo de producción (comisiones, agentes de seguros, comisiones descontado, etc.)

Los gastos de gestión externa se producen mientras existan pagos de primas y es un porcentaje de todas las primas comerciales, y su porcentaje se lo representa como b .

Por el contrario los gastos de gestión interna se produce mientras la póliza siga vigente y la forma de establecerlo es un porcentaje del capital asegurado, este se lo representa por g .

Los gastos de producción son un porcentaje de la prima comercial y su porcentaje se representa por a .

Ejemplo

Para un seguro de vida entera deje expresado las obligaciones anuales del asegurado, en términos de la prima de inventario, comercial, y de Zillmer.

Prima de Inventario.

$$P'_x \ddot{a}_x = A_x + g \ddot{a}_x \Rightarrow P'_x = \frac{A_x}{\ddot{a}_x} + g \quad (3.137)$$

Prima Comercial

$$P''_x \ddot{a}_x = A_x + g \ddot{a}_x + \mathbf{b} \ddot{a}_x P''_x + \mathbf{a} P''_x \Rightarrow P''_x = \frac{A_x + g \ddot{a}_x}{\ddot{a}_x - \mathbf{a} - \mathbf{b} \ddot{a}_x} \quad (3.138)$$

Prima de Zillmer

$$P_x^Z \ddot{a}_x = A_x + aP_x'' \Rightarrow P_x^Z = \frac{A_x}{\ddot{a}_x} + a \frac{P_x''}{\ddot{a}_x} \quad (3.139)$$

3.6. Reservas Matemáticas

En una operación de seguros con pagos de primas anuales netas al momento de concertar la operación se verifica el principio de equivalencia actuarial, es decir, el valor actuarial de las primas futuras es igual a las prestaciones futuras por parte del asegurador, por lo que la pérdida esperada es nula, pero los seguros tienen una vigencia de larga duración y la equivalencia entre pagos futuros y prestaciones futuras por lo general no existe a lo largo del tiempo, debido a esto la aseguradora deberá mantener en sus pasivos recursos monetarios igual o mayor a esta diferencia, para cubrirse de esta situación en un tiempo t dado, este monto es lo que se denomina la Reserva matemática, El siguiente ejemplo ilustrara lo expuesto.

Una aseguradora esta planeando emitir una póliza de vida entera para un recién nacido donde la variable aleatoria K está gobernada por la siguiente función de probabilidad.

$${}_k|q_0 \quad K = 0, 1, 2, 3$$

- a) Determine la prima anual neta donde el valor esperado del valor presente, de la perdida financiera sea igual a cero.
- b) Evalúe el valor presente de las obligaciones futuras financieras al tiempo 1, Utilizando la prima anual encontrada en (a)

Con una tasa del 6%

Solución:

Literal a

De acuerdo el literal a) el valor de P es tal que

$$\sum_{k=0}^3 \left(v^{t+1} - P \ddot{a}_{\overline{k+1}|} \right) \Pr(K = k) = 0$$

Lo cual nos da un $P = 0.3667$

Literal b

Tenemos que la función de probabilidad condicional de K tal que $K \geq 1$ es

$$\Pr(K = k / K \geq 1) = \frac{\Pr(K = k)}{\Pr(K \geq 1)} = \frac{1/4}{3/4} = \frac{1}{3} \quad K = 1, 2, 3$$

Utilizando la prima calculada en el literal a) como las obligaciones anuales de asegurado tenemos la siguiente tabla

Tabla II

Valor presente de las obligaciones futuras de

Año	Probabilidad	Asegurado	Aseguradora	Perdida
1	1/3	0.9434	0.3667	0.5767
2	1/3	0.8900	0.7126	0.1774
3	1/3	0.8396	1.0390	-0.1994

Tenemos que el valor presente de la pérdida de la aseguradora es igual a:

$$\frac{1}{3}(0.5767+0.1774-0.1994)=0.1849$$

Este valor es lo que la aseguradora debe tener como reserva para resguardarse de posibles pérdidas.

De igual manera sea j la variable aleatoria tiempo futuro de vida abreviado en $(x+k)$, la variable de pérdida en este punto es:

$${}_xL = v^{j+1} - P_x \ddot{a}_{\overline{j+1}|} \quad (3.140)$$

El valor esperado de esta variable es lo que se denomina la reserva matemática en el punto $x+k$ y se la expresa de la siguiente manera.

$${}_kV_x = E({}_kL) = E(v^{j+1}) - P_x E\left(\ddot{a}_{\overline{j+1}|}\right) \quad (3.141)$$

La reserva matemática de un individuo (x) en la edad (x+k) para una operación de seguro de vida entera de un capital unitario pagadero al final del año de fallecimiento, y donde las primas se cancela anualmente mientras sobreviva el asegurado, se puede expresar de la siguiente forma.

$${}_kV_x = E({}_kL) = E(v^{j+1}) - P_x E\left(\ddot{a}_{j+1}\right) = A_{x+k} - P_x \ddot{a}_{x+k} \quad (3.142)$$

De esta forma se pueden calcular las reservas para cualquier tipo de planes, como los siguientes.

3.6.1. Reserva matemática para un plan compuesto de una operación de seguros para un individuo (x) diferido a m años y con vigencia a n años con pagos de primas mientras viva a lo sumo m años $m \leq n$

La prima neta pura es

$$P_{x:n} = \frac{A_{x:n}^1}{\ddot{a}_x} \quad (3.143)$$

Para calcular la reserva tenemos que analizar por cuantas situaciones distintas se desplaza k en el transcurso del tiempo, así tenemos los siguientes casos:



La situación que se tiene en el primer caso es que tanto la aseguradora como el asegurado tienen obligaciones entre sí, no así en el caso 2 donde ya el asegurado si es que ha sobrevivido a la edad $x+m$ ya no tiene obligaciones pero la aseguradora aún tiene alguna probabilidad de cancelar la indemnización, en el último caso ni la aseguradora, ni el asegurador tienen obligaciones, el contrato ha finalizado.

Así tenemos que la reserva para este plan es:

$${}_k V_{x:n} = \begin{cases} A_{x+k:n-k}^1 & k > 0 \text{ y } k < m \\ A_{x+k:n-k}^1 - P_{x:n} \ddot{a}_{x+k:m-k} & k > m \text{ y } k < n \\ 0 & k > n \end{cases}$$

CAPITULO IV

4. HERRAMIENTAS INFORMÁTICAS PARA EL DESARROLLO DE LA HOJA ELECTRÓNICA

La herramienta escogida para el desarrollo de la Hoja electrónica actuarial es el lenguaje de programación basado en objetos Visual Basic 6.0, donde la palabra "Visual" hace referencia al método que se utiliza para crear la interfaz gráfica del usuario (GUI), la palabra "Basic" hace referencia al lenguaje BASIC, ya que Visual Basic es la evolución de este lenguaje con más herramientas que están directamente relacionada con la interfaz gráfica de Windows.

Visual Basic es un lenguaje de programación que utiliza componentes preestablecidos, pero desde la versión 4 cuando se incorporo dentro de las herramientas de programación el módulo de clase éste fue capaz de crear objetos personalizados por el usuario, pero aún así no es posible decir que VB es un lenguaje netamente orientado a objetos, ya que no cumple con ciertas características para ser catalogado de esta manera, como la herencia, pero estos módulos de clase si posibilitan la implementación de algunas características claves orientadas a objetos. Una de la más importante es la de encapsular la información tras una interfaz claramente definida.

4.1. Conceptos claves

4.1.1. Tiempo de diseño

Este es el tiempo donde se bosqueja, en el entorno de desarrollo la interfaz gráfica de la aplicación y se describe el comportamiento de los controles por medio de sus eventos y métodos o se les asigna valores a sus propiedades.

4.1.2. Tiempo de ejecución

En este tiempo la aplicación se encuentra ejecutándose y permite interactuar con ésta de la misma manera que el usuario lo haría en una ejecución real.

4.1.3. Interfaz de un solo documentos SDI

Este tipo de interfaz permite una o más ventanas abiertas, pero cada una de ellas puede ser minimizadas o maximizadas independientemente una de la otra.

4.1.4. Interfaz de múltiples documento MDI

Si una aplicación admite una interfaz MDI es por que posee una sola ventana principal y varias ventanas contenidas en ésta, cada vez que se cierra la ventana principal las demás ventanas secundarias efectúan el mismo evento.

4.1.5. Entorno integrado de desarrollo (IDE)

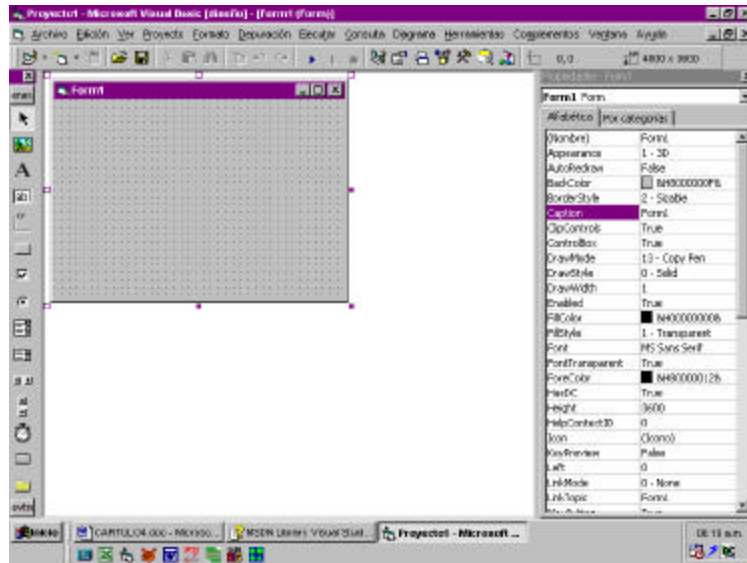


Figura 4.1. Entorno integrado de desarrollo

Es el ambiente donde se desarrollan las aplicaciones, brindando las herramientas necesarias para lograr este fin, de una manera eficiente y cómoda para el usuario de Visual Basic, entre algunas de estas tenemos el cuadro de herramientas donde se encuentran todos los componentes activados que pueden formar parte de la interfaz visual de una aplicación, el examinador de objetos brinda la posibilidad de tener un control de todas las

propiedades y métodos de los objetos existentes en la aplicación, el explorador de proyectos permite observar todos los proyectos que conforman una aplicación así como las formas, los módulos estándar, y los módulos de clase que contiene cada proyecto.

4.1.6. Interfaz de un objeto

En Visual Basic un objeto puede poseer dos tipos de interfaz, la visual que interactúa directamente con el usuario del sistema, el cual permite observar al objeto gráficamente y así en tiempo de diseño ubicarlo en la posición deseada y con el tamaño requerido, la otra interfaz interactúa en forma directa con la aplicación cliente o el contenedor del objeto, exponiéndole a este sus métodos eventos y propiedades, es entonces ésta interfaz que utiliza el programador para darle a un objeto un comportamiento específico.

No es estricto que un objeto posea interfaz visual, puede poseerla o no, es el caso de los objetos creados desde las DLL Active X que no poseen una interfaz visual, pero si es necesario que posea propiedades, métodos y eventos que interactúen con la aplicación cliente.

4.1.7. Código reutilizable

El concepto de código reutilizable es muy importante en la programación orientada a objetos, y está muy ligado a la propiedad de encapsulamiento que un objeto debe poseer, esto es que a la aplicación que se sirve de un objeto no le interesa cómo este realiza sus procesos internos mientras los resultados que arrojen sus interfaces sean los adecuados así como sus parámetros.

4.1.8 Controles estándar de Visual Basic



Figura 4.2. Controles estándar

Los controles estándar son aquellos que son parte de visual Basic y no representan un recargo adicional por licencia para su utilización, estos controles son los más comúnmente usados en una aplicación.

4.1.10. Cuadro de Referencias

Este permite agregar al proyecto una biblioteca de objetos, una biblioteca de tipos o una referencia de proyecto, esto hace disponible todos los objetos de una aplicación, así por ejemplo un proyecto puede hacer uso de características de un procesador de palabras como Word o de una hoja electrónica como Excel.

4.1 Ámbito de las variables VB

El ámbito de una variable hace referencia al alcance que tienen éstas dentro de un proyecto de Visual Basic, este alcance varía según su tipo.

Las funciones al igual que las variables pueden ser públicas, la versión Dim para funciones es Private, existe un tercer tipo de alcance denominado Friend utilizado con frecuencia en los Componentes Active X, ésta será tratada más adelante en este capítulo.

4.2.1 Declaración Dim

Una variable se la declara Dim dentro de un procedimiento o función, su alcance sólo llega dentro de las mismas, es decir estas variables solo son visibles dentro de estos procedimientos y funciones.

4.2.2 Declaración Public

Una variable de este tipo posee un alcance global, es decir puede ser utilizada en cualquier parte del sistema, generalmente ésta variable se la declara en la sección general de los módulos estándar.

4.2.3. Declaración Static

Una variable declarada de este tipo tiene un alcance local igual que las variables Dim pero con la diferencia que ésta no pierde su valor final al culminar la función o procedimiento donde fue declarada.

4.3 Módulos estándar y módulo de clase

Tanto los módulos estándar como los de clase son segmentos de código donde se declaran funciones, variables y métodos para la posterior utilización de éstos en el programa principal.

Existe una gran diferencia entre ambos módulos, básicamente en la forma en que almacenan los datos, por ejemplo dos secciones de un programa pueden acceder a una variable declarada en un módulo estándar como pública, la una modifica la variable, y la otra podrá leer la misma variable ya modificada, por el contrario los datos de un módulo de clase existen para cada instancia de la clase. Es decir existirán tantas versiones de una variable declarada en un módulo de clase como instancia tenga el objeto, cada una independiente de la otra.

4.4. Creación de objetos personalizados en Visual Basic

La interfaz de un módulo de clase es el medio por el cual un objeto se comunica con el "Mundo Exterior", esto es la aplicación o sector

de código que lo invoca, esta interfaz se compone de funciones, sub procedimientos y propiedades, estos dos últimos componentes en la terminología de la programación orientada a objetos se los denominan métodos del objeto.

Visual Basic al generar módulos de clases brinda la posibilidad al usuario de generar sus propios objetos personalizados, e interactuar entre objetos mediante el polimorfismo, la hoja electrónica actuarial desarrollada está basada en objetos personalizados.

4.4.1. Componentes de un módulo de clases

4.4.1.1. Propiedades

Una propiedad es una característica específica de un objeto, ésta se implementa internamente por dos métodos, una para establecer la propiedad (Property Let) y otra para recuperar el valor de la misma (Property Get).

4.4.1.2. Métodos de una clase

Un método como ya se mencionó es una función o procedimiento, el cual representa una acción del objeto, esta función puede o no devolver un valor.

4.4.2. Colecciones de objetos

Una colección es una agrupación de objetos relacionados entre sí. En Visual Basic existen ya colecciones predeterminadas como la colección Form, los objetos que contiene ésta colección son todos los controles que tienen una instancia en la forma.

Visual Basic también brinda posibilidad de crear colecciones propias mediante el objeto COLLECTION, por medio de éste podemos mantener colecciones propias de objetos personalizados.

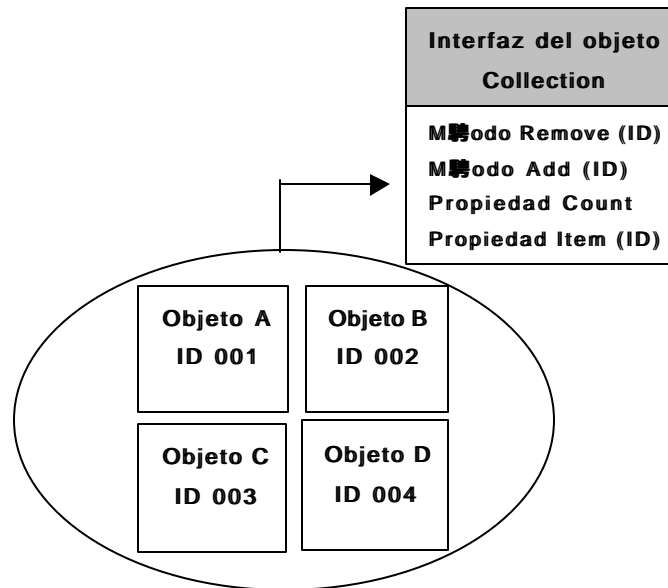


Figura 4.3. Estructura de un objeto colección

La interfaz del objeto Collection está compuesta por dos propiedades y dos métodos como se observa en la figura 4.3 los cuales permiten tenerlos organizados dentro de la colección, entre estos tenemos el método “Add”, el cual añade un nuevo elemento a la colección, así tenemos también su contraparte el método “Remove”. La propiedad “Count” nos ofrece el número de elementos existentes en una colección, la propiedad Item nos devuelve el objeto requerido por medio de una clave que lo identifica dentro de la colección, con el fin de manipular alguna propiedad o método de este objeto.

4.4.3. El modelo de objetos

Los modelos de objetos aportan la estructura a los programas basados en objetos, nos brinda una oportunidad de ordenarlos para facilitar la codificación.

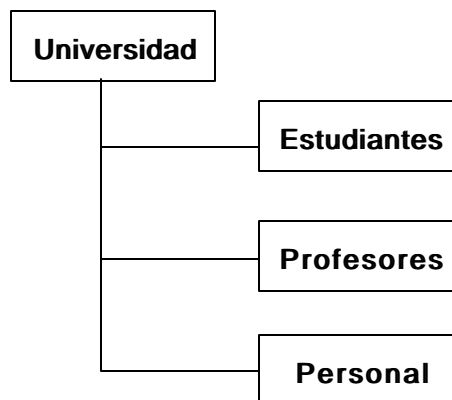


Figura 4.4. Modelo de objetos para una universidad

Un modelo de objeto diferencia entre objetos grandes y pequeños, u objetos contenedores y contenidos relacionándolos de manera directa, cómo se observa en la figura 4.4, el objeto Universidad es un objeto contenedor de los objetos Estudiantes, Profesores y Personal.

La creación de un modelo de objetos en Visual Basic es sencilla, sólo se crea una variable de la clase contenida dentro de la clase contenedora y se las presenta por medio de propiedades de lectura y escritura (Property Set , Property get) es decir que los objetos contenidos forman parte de la interfaz del objeto contenedor.

4.4.4. Vida interna de un objeto

4.4.4.1. Variable de objeto

Una variable de objeto no almacena el objeto como tal, ni tampoco datos como lo hacen las variables normales si no una referencia hacia una de las interfaces del objeto, por lo tanto cada variable de objeto solo toma 4 bytes de memoria debido a que no es nada más que un puntero a una dirección de memoria.

Uno de los resultados interesantes de lo expresado en el párrafo anterior resulta cuando varias variables apuntan a la misma instancia del objeto, se pueden manipular las propiedades del objeto usando cualquiera de las variables porque todas ellas apuntan a la misma área de datos.

4.4.4.2. La palabra clave new

Esta palabra clave indica a Visual Basic que debe de crear una nueva instancia de una clase dada. La palabra clave retorna la dirección de memoria del área de datos de la instancia recién creada.

4.4.4.3. La palabra clave set

Esta palabra clave indica a que dirección de memoria debe de apuntar un objeto, es el equivalente a la asignación de variables.

4.4.4.4. La palabra clave Nothing

Con esta palabra reservada de visual se anula un objeto de memoria, hace que el puntero del objeto apunte a Null.

4.4.4.5. La palabra clave Is

Esta palabra reservada de Visual Basic nos indica si dos variables de objeto están apuntado a la misma dirección de memoria.

4.4.4.6. La palabra clave typeof

Nos indica de que tipo es un objeto, o de que clase fue instanciada una variable de objeto.

4.4.4.7. La función TypeName

Esta palabra clave retorna el nombre de la clase de donde fue instanciada el objeto.

4.4.4.8. El evento Class_Initialize

Este evento se ejecuta en el momento de crear una instancia de la clase, en esta sección se puede escribir código de inicio del objeto o inicializar variables.

4.4.4.9. El evento Class_Terminate

Este evento se ejecuta un instante antes de liberar una instancia de la clase y terminar la vida del objeto. Generalmente se escribe código cuando se necesita deshacer cosas que se hicieron en el momento de inicializar o durante la vida del objeto.

4.5. Acceso a datos

4.5.1. Conectividad ODBC y OLEDB

ODBC es un componente desarrollado con el fin de resolver las incompatibilidades entre distintas bases de datos para aplicaciones cliente / servidor, ya que cada fabricante de base de datos tenía un API particular. ODBC soluciona este problema mediante la definición de una API del lado del cliente, el cual es independiente de la base de datos que haya en el otro extremo.

OLEDB es muy parecido a ODBC, pero diferente en algunos aspectos, por ejemplo OLEDB esta basado en el Modelo de Objeto Componente (COM), una arquitectura muy robusta utilizada en la actualidad, segundo OLEDB se presta para establecer conexiones con distintos tipos de fuentes de datos, no sólo con datos ISAM (Index Sequential Acces Mode), lo cual es el campo natural de ODBC.

4.5.2. La arquitectura ActiveX Data Object (ADO)

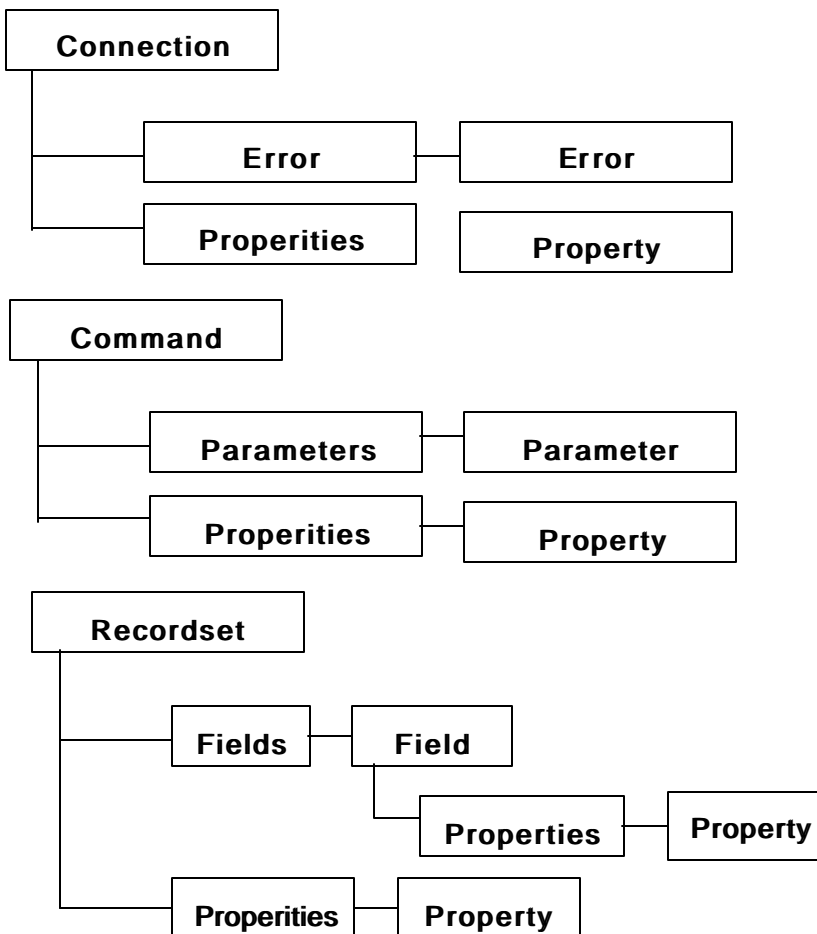


Figura 4.5. Modelo de objeto ADO

Esta arquitectura ofrece los objetos necesarios para el acceso y manipulación de datos almacenados en alguna fuente. A parte de ADO existen otras arquitecturas de acceso a datos como el DAO o RDO, pero ADO es relativamente más simple que las dos

anteriores puesto que este tiene mucho menos objetos y colecciones.

Como se observa en la figura 4.5 el modelo de objetos completo de ADO, este tiene tres objetos independientes, el objeto connection, el objeto recordset, y el objeto command, cada una de las cuales exponen dos colecciones.

4.5.2.1. El objeto connection

El objeto connection representa una puerta de conexión hacia un origen de datos, este origen puede ser una base de datos, un origen ODBC o cualquier otro origen donde exista un proveedor de OLEDB.

El objeto connection especifica todos los parámetros necesarios cómo por ejemplo el nombre del usuario, del servidor o el nombre de la base de datos.

4.5.2.2. El Objeto Recordset

Este objeto contiene todos los datos que se lee de una base de datos, o aquellos datos que se quieren enviar a la base, posee varias filas y columnas donde cada fila es un registro, cada columna es un campo. Aparte de poder contener una tabla entera de una base de datos, también puede ser el resultado de una consulta, combinando dos o más tablas por medio de sentencias Sql.

El Objeto recordset de DAO es mucho más versátil que el de ADO y RDO puesto que se puede crear un objeto recordset sin establecer una conexión a una base u obtener un objeto recordset de una base de datos, cerrada la conexión, alterar los datos en el recordset, y finalmente restablecer la conexión para enviar todos los datos modificados.

4.5.2.3. El objeto Command

Este objeto define un comando o consulta la cual puede ser ejecutada sobre un origen de datos. El objeto Command es muy

útil cuando se planea ejecutar el mismo comando o consulta varias veces (sobre la misma o diferentes orígenes de datos).

4.6. DLL Active X

4.6.1. El camino hacia Active X

4.6.1.1 Intercambio dinámico de datos

La tecnología Active X no es reciente, este ha experimentado un proceso de transformación a través del tiempo, el inicio de esta tecnología fue DDE (Intercambio de datos dinámicos). Esta todavía es admitida por algunas aplicaciones de Visual Basic, pero de manera general existen muy pocas aplicaciones que soporten DDE.

Esta tecnología ofrecía medios para permitir que las aplicaciones se comuniquen entre sí, los datos venían identificados con el nombre de la aplicación, por el tema individual.

4.6.1.2. Vinculación e Incrustación de objetos 1.0

El siguiente paso de la tecnología Active X fue la Vinculación e Incrustación de objetos 1.0 (OLE 1.0), esta tecnología dio un gran paso en uno de los conceptos más importantes de la computación centrada en datos, esto es que un documento soporte distintos tipos de objetos, de ahí el nombre de incrustación y vinculación de objetos.

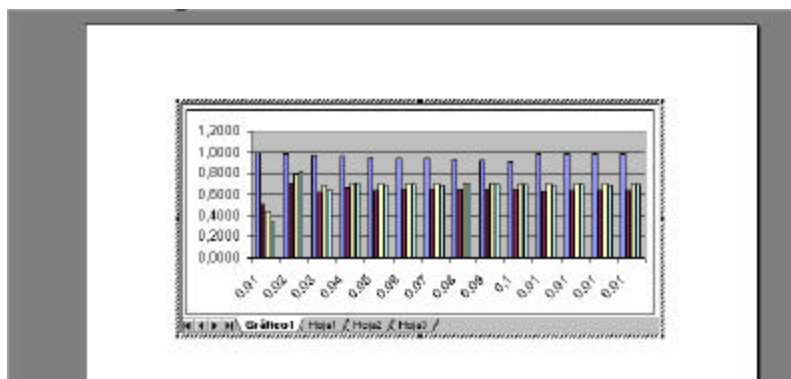


Figura 4.6. Gráfico de Excel incrustado en un documento de Word

Un documento podría usar distintos tipos de objetos asociados a aplicaciones distintas y editarlos dentro de las mismas aplicación de donde provienen, esto se puede observar en la figura 4.6, donde se observa un documento de Word el cual tiene incrustado

un gráfico de Excel. Al darle doble clic sobre el objeto gráfico para editarlo, se iniciará la aplicación de Excel.

4.6.1.3. Vinculación e Incrustación de Objetos 2.0

La siguiente versión de OLE brinda la posibilidad de que un objeto sea incrustado, modificado, guardado en una aplicación sin que ésta supiera acerca del objeto, inclusive esto debe funcionar para aquellos objetos que se desarrollen después de la creación de la aplicación.

Para manejar la posibilidad antes planteada OLE 2.0 propone que los objetos que se creen en este contexto sigan un estándar llamado el Modelo de Objetos Componente (COM).

4.6.1.4. La tecnología Active X

Active X es la estrategia comercial de Microsoft debido a la aparición de internet y a la necesidad de acoplarse a la red, Microsoft necesitaba dar a conocer al mundo que tenía una propuesta seria para esto, así que rebautizó OLE como Active X,

entonces OLE es Active X, todos los controles Active X son controles OLE no necesariamente orientados a Internet.

4.6.1.5. identificadores de los objetos

Una de las características de los objetos generados en OLE es la identificación que estos tienen para que el sistema pueda reconocer la aplicación que maneja el objeto. Para alcanzar este fin COM asigna a cada objeto un valor de 16 bytes, este identificador toma distintos nombres dependiendo de cómo se utilice, así tenemos que HUID es el identificador Único Universal, GUID identificador Único Global, CLSID identificador de Clase, mientras que IID significa identificador de Interfaz.

El identificador GUID es único universalmente es decir no existen en el mundo dos objetos con el mismo identificador, para conseguir esto, parte del GUID se genera de la dirección única de la tarjeta de red de cada PC, además si no existiera la tarjeta de red en la PC un identificador GUID es muy grande por lo tanto la probabilidad de que ocurra un duplicado es muy pequeña.

4.6.2. El modelo de objeto componente

El modelo objeto componente (COM) define cuáles son las características que un objeto debe poseer para que aplicaciones diferentes de donde se generó el objeto lo manipulen.

4.6.2.1. Características de los objetos COM

El modelo de objetos componentes ofrece normas para el manejo óptimo de los objetos dentro de la tecnología Active X, estas normas establecen características específicas, de las que se pueden mencionar. Las interfaces las cuales se las pueden llamar para actuar en los datos, o llevar a cabo otras operaciones como la persistencia de datos.

Las funciones del objeto están contenidas dentro de librerías DLL, estas son las únicas funciones que pueden entrar a los datos del objeto.

Los objetos COM esta identificado por un GUID, el cual lo identifica de forma única en el mundo, el GUID se lo utiliza para asociar una determinada instancia de un objeto con la aplicación que lo invoca.

Un objeto COM puede exponer un sin número de interfaces, que le permitan dibujarse en un contexto de dispositivo, o guardarse en un archivo o flujo por medio de almacenamiento estructurado OLE, no todos los objetos COM exponen todas estas interfaces, pero todo objeto COM debe implementar por lo menos una interfaz especial denominada IUnknown, ésta se compone de tres funciones AddRef, Release, Query Interface.

AddRef funciona cómo un contador de referencias, es decir cada vez que se realice una referencia a un objeto se invoca la función AddRef aumentando el contador, Release funciona de un modo contrario a AddRef es decir cada vez que se establece un objeto a NULL se invoca a esta función decrementando el contador de referencia. Query Interfaz es algo más complejo puesto que es la encargada de preguntar si un objeto soporta o no una determinada interfaz, y si lo hace obtener un señalizador hacia la interfaz requerida.

4.6.2.2. Interfaces de Automatización o Envío

Un objeto COM debe dar la oportunidad de definir funciones y propiedades personalizadas para así exponerlas en una lista con

información de sus parámetros e identificadores respectivos, para que otros objetos puedan acceder a esta información e invocar cualquier función o método disponible en la lista, las interfaces de automatización proporcionan una metodología para que esto ocurra.

COM hace uso de la interfaz IDispatch para lograr el objetivo planteado en el párrafo anterior, la interfaz delega tareas específicas a cuatro funciones que la componen estas son la interfaz GetTypeInfoCount, GetTypeInfo, GetIDsOfNames y la función Invoke.

GetTypeInfoCount y GetTypeInfo se utilizan para mostrar una lista de funciones de las funciones y métodos, la información incluye los nombres de las funciones, los parámetros, tipos de parámetros y valores de retorno.

La función GetIDsOfName permite obtener el identificador de un método o propiedad dado el nombre de ésta, así como los tipos de parámetros de la propiedad o método.

La función Invoke invoca a una función o método, tomando el identificador especificado y una matriz de variantes y estructuras que contienen información sobre los parámetros.

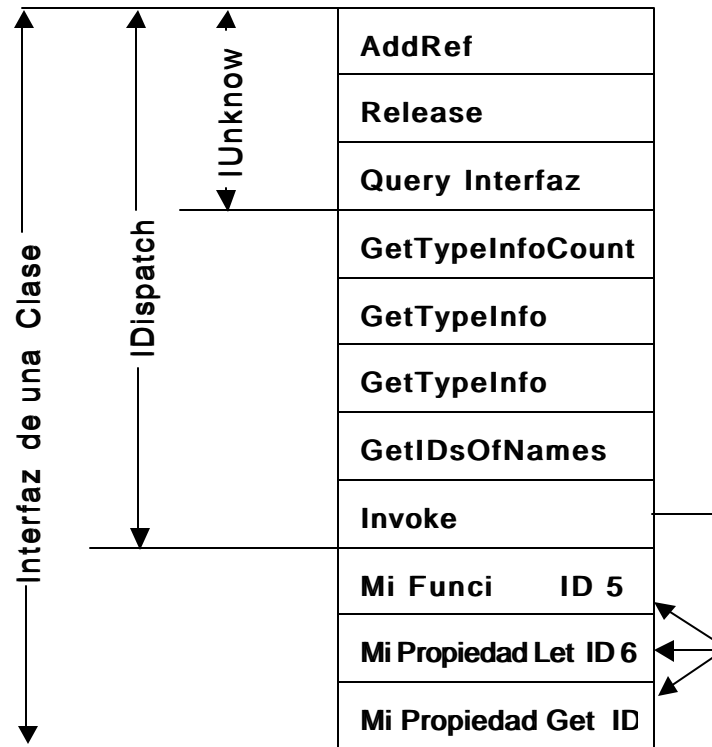


Figura 4.7. Estructura de un módulo de clase

La figura 4.7 muestra la relación existente entre las interfaces IUnknown y la IDispatch ambas interactúan cuando VB necesita acceder a los métodos e interfaces de un objeto. Query Interface obtiene una referencia a la interfaz IDispatch del objeto. Luego llama a la función Invoke para que ejecute el método "Mi Función"

o a la propiedad "Mi Propiedad", este proceso ocurre cuando el objeto es declarado como "as Object", pero cuando VB sabe con antelación la clase de donde proviene el objeto este usa solo Query Interfaz para obtener una referencia a la clase para así llamar a la función "Mi Función" o a la propiedad "Mi Propiedad" directamente sin llamar a la función Invoke.

Al ver la clase expuesta en la figura 4.7 se aprecia dos tipos de funciones las creadas por el programador y las de automatización, de estas últimas nunca el programador se va a preocupar de desarrollarlas puesto que VB se encarga de ello y con esto toda la complejidad que esta interfaces conllevan.

4.6.3. Tipos de controles y componentes Active X

VB al estar basado en COM, todos los objetos creados en este lenguaje poseen las características ya mencionadas de los objetos COM. Un componente Active X es una unidad de código ejecutable que proporciona funcionalidad, donde los objetos generados por este componente son objetos OLE.

Los componentes los proporciona los servidores .exe, .dll, u .ocx. COM brinda especificaciones para que una aplicación cliente se conecte a un servidor y este pueda brindar al cliente los objetos que genera.

Existen varios tipos de componentes Active X que VB puede crear, los componentes de código Active X , controles Active X y documentos Active X. Lo que resta de este capítulo vamos a detallar sólo los componentes de código Active X puesto que la hoja electrónica actuarial se componen de estos.

4.6.4. Componentes de Código Active X

Los componentes de código Active X se desglosa en las DLL Active X o servidores dentro de proceso y componentes EXE Active X o también conocidos cómo servidores fuera de proceso.

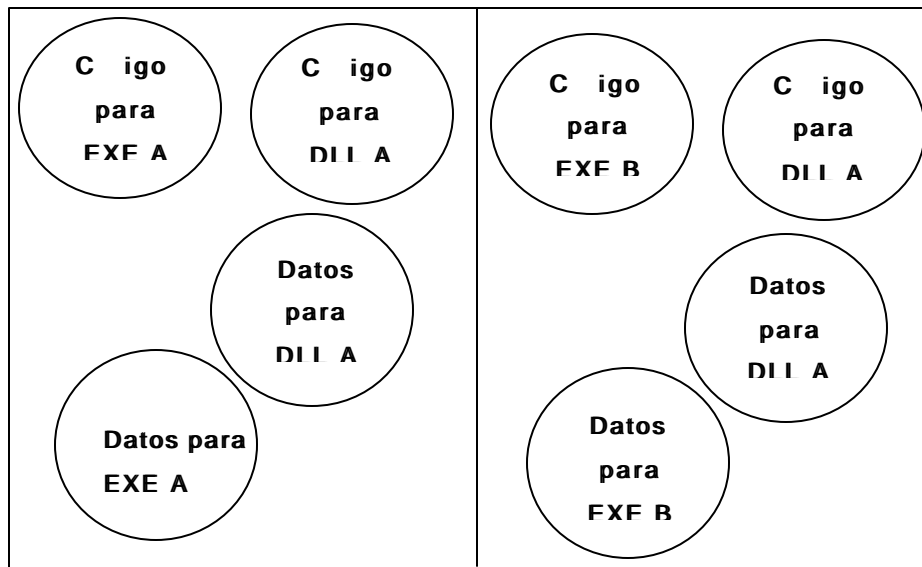


Figura 4.8. Organización de la memoria con dos aplicaciones ejecutándose.

Win32 utiliza una arquitectura de memoria como la esquematizada en la fig 4.8, donde tenemos dos aplicaciones ejecutándose al mismo tiempo los cuales hacen referencia a una DLL A, Win32 a cada aplicación le asigna su propio espacio de proceso y el sistema operativo divide el tiempo disponible de la CPU entre los procesos en ejecución.

Como se observa en la figura la DLL A se ejecuta en el mismo espacio de proceso de la aplicación a la que esta referida por lo

que el acceso a las interfaces de este componente es sumamente rápido comparado con los EXE Active X, de ahí el nombre de las DLL Active X como servidores dentro del proceso. Todas aquellas aplicaciones que admitan Automatización OLE pueden hacer referencia a las DLL Active X creadas en VB

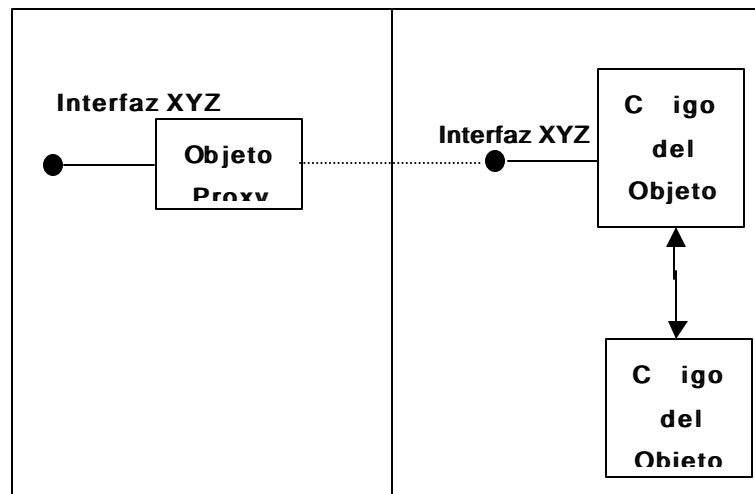


Figura 4.9. Esquema del llamado de una función contenida en un componente EXE Active X

Cuando una aplicación requiere una función de un componentes EXE Active X este debe de pasar su propio espacio de proceso e ingresar al espacio de proceso donde se encuentra los objetos requeridos, pero esto no es así de sencillo, ya que debido a la separación que hay entre espacios de procesos, no es posible

llamar directamente a funciones que pertenezcan a otro proceso. Pero OLE resuelve este problema generando un objeto Proxy como se observa en la figura 4.9 este objeto exhibe la misma interfaz que el verdadero objeto, de ahí el nombre de los componentes EXE Active X como Servidores fuera de proceso.

Los componentes EXE Active X generalmente se crea cuando se requiere que una aplicación trabaje por si sola y a su vez ofrecer objetos programables al exterior al mismo tiempo, existen ejemplo de servidores EXE Active X que los usuarios Office se topan con frecuencia como Microsoft Word, Microsoft Excel, Mocosoft Power Point etc.

4.6.5. Ventajas de las DLL Active X y EXE Active X

Una ventaja de utilizar una DLL es el rendimiento, puesto que este se ejecuta dentro del proceso de la aplicación que la invoca.

Las DLL pueden ser utilizadas por clientes que admitan automatización OLE entre las que se incluye las aplicaciones basadas en VBA como Microsoft Office.

Las DLL pueden ser utilizadas por más de un cliente, puesto que permite compartir código entre las aplicaciones.

Los objetos de los componentes EXE Active X se pueden ejecutar en su propio subproceso, además pueden ser creados y utilizados tanto por aplicaciones cliente como aplicación autónoma.

4.6.6. Desventajas de las DLL Active X y EXE Active X

Una de las desventajas de las DLL es que si está actualizada es incompatible con su predecesor, podrá dividir la aplicación que emplee la DLL. Además requiere del registro, la comprobación de versiones y la verificación de componentes para una distribución más segura, esta desventaja también la posee un componente EXE Active X.

Los componentes EXE Active X presentan un menor rendimiento que las DLL puesto que para acceder a alguna función se necesita crear un objeto Proxy. Además aumenta la complejidad de desplegar una aplicación.

CAPITULO V

5. DISEÑO DE LA HOJA ELECTRÓNICA ACTUARIAL

Para el diseño de Maxim nos fundamentamos en los pasos del Ciclo de Vida del Desarrollo de un Sistema (o sus siglas en ingles SDLC) los cuales a breves rasgos esta conformado por 1) El análisis y diseño del sistema 2) El desarrollo del sistema 3) Implementación y Pruebas, cada uno de estos no son independientes entre si, si no que su relación es directa.

De los cuatro pasos mencionados, se derivan sub etapas donde se desarrollan los documentos necesarios para el soporte de cada paso, estas etapas están desglosadas de la siguiente forma.

Análisis y Diseño

- ✓ Identificación de oportunidades y objetivos
- ✓ Determinación de los requerimientos de información
- ✓ Análisis de las necesidades del sistema
- ✓ Diseño del Sistema

Desarrollo

- ✓ Desarrollo y documentación del software

Implementación y Pruebas

- ✓ Pruebas y mantenimiento del sistema
- ✓ Implantación y evaluación del sistema

El presente capítulo engloba las etapas de análisis y diseño con sus correspondientes sub etapas, las restantes serán cubiertas en el capítulo 6.

5.1. Análisis

5.1.1. Identificación de Inconvenientes y oportunidades

En el mercado existen productos informáticos orientados netamente al análisis estadístico de la información entre estos se pueden mencionar Systat, SPSS, SPLUS, etc y aquellos donde con un poco de esfuerzo dan la posibilidad de combinar funciones estadísticas y financieras básicas, de este grupo se destaca Excel de Microsoft. Pero es muy dificultoso encontrar un producto que este realmente orientado al análisis y cálculo actuarial, que permita tanto al actuario, como al estudiante de la ciencia actuarial el análisis de modelos específicos, y más aún la comparación entre dos o más modelos actuariales entre otras posibilidades.

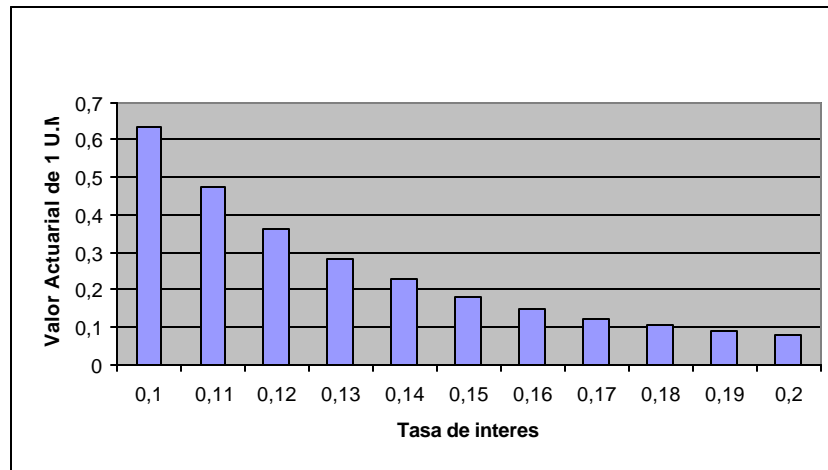
Para varias compañías de seguros en el Ecuador la herramienta informática predeterminada para el análisis actuarial es la hoja de calculo de Microsoft, pero esta no presenta las prestaciones

actuariales ideales, y como se mencionó esta más orientado al análisis financiero contable.

También es cierto que el ICM no posee una herramienta informática sólida, que de la posibilidad a sus estudiantes de actuariales una mayor comprensión del comportamiento de los modelos, al afrontarse a la variación de sus parámetros para distintas leyes y tablas de mortalidad. Y un paquete informático de este tipo vendría a ser una invaluable ayuda didáctica y profesional.

Los modelos actuariales están compuestos por diversos elementos, entre los que se tienen la tasa de interés nominal, la vigencia, la edad del asegurado etc. Las variaciones de algunos de estos parámetros sensibilizan, unos más que otros el valor actuarial del modelo o al riesgo, por ejemplo se sabe que, uno de los parámetros que más afecta el valor actuarial en la mayor parte de los modelos ocurre cuando la tasa de interés varía en porcentajes pequeños, como se observa en el gráfico siguiente, para un cambio del 1% a una tasa inicial del 10%, para un seguro de vida completa de una unidad monetaria, para una persona de

25 años de edad, utilizando la tabla ecuatoriana de mortalidad del año de 1990.



Fuente : Tablas de mortalidad de 1990

Figura 5.1. Valor Actuarial para diferentes tasas de interés

La Figura 5.1 nos deja ver el cambio gradual del valor actuarial, con una variación de un punto porcentual en la tasa de interés. Con una tasa del 10% se exige un poco más del 60% de la indemnización como prima pura, mientras que con 20% de interés se exige un poco menos del 10% como prima pura de la indemnización total, entonces se concluye que con una variación del 10 al 20 % en la tasa de interés, la variación en el valor

actuarial es de 60 al 10 % de la indemnización total, realmente esto es considerable.

Las hojas electrónicas comunes, orientadas más al análisis contable-financiero, no presentan la facilidad y flexibilidad que se requiere para realizar estos tipos de cálculos y el análisis correspondiente para los distintos modelos de seguro y renta de una manera eficiente.

Es lógico suponer que las leyes de mortalidad que gobiernan dos o más conglomerados son diferentes entre sí, por lo tanto el análisis de las características de estas leyes y la manera en que influyen en los cálculos actuariales es de sumo interés para la toma de decisiones del actuario o la comprensión del estudiante de la ciencia actuarial.

La falta de una cultura actuarial, que no ha sido establecida entre los organismos responsables, por la carencia de información e iniciativa para el desarrollo de tablas diversificadas, ha influido en un bajo nivel de investigación y análisis de nuestra realidad nacional actuarial. Un ejemplo latente en nuestro país es el cálculo y análisis de primas, o reservas con tablas construidas en países

vecinos como Colombia o Perú. Esto presenta un Inconveniente para cualquier herramienta informática orientada al cálculo y análisis de los modelos actuariales, donde una de las principales entradas al sistema son las tablas de mortalidad.

5.1.2. Definición del sistema

Maxim es el nombre de la aplicación, que está orientada al cálculo de los valores actuariales y análisis de los diferentes modelos de seguros y rentas continuos o discretos, asignados a una persona de “X” años de edad.

Es una aplicación de escritorio, monousuario cuya interfaz visual presenta características de una hoja electrónica como Excel, ideal para la manipulación de datos numéricos.

Los cálculos actuariales que requiere Maxim, son efectuados por códigos que están establecidos en componentes DLL Active X, denominados Seguros.dll y Rentas.dll, las características de los modelos se encuentran organizados en clases y cada vez que se requiera un modelo específico se crea una instancia de la clase del modelo. Maxim hace las veces de cliente, y los componentes DLL

de servidores de modelos para Maxim, éste los utiliza con el fin de calcular las distintas primas, y reservas matemáticas.

Para el diseño del sistema se piensa en el usuario como parte de éste, se considera iteraciones entre ambos, formado un ciclo de información que va desde el sistema al usuario y viceversa, como se observa en el gráfico siguiente.

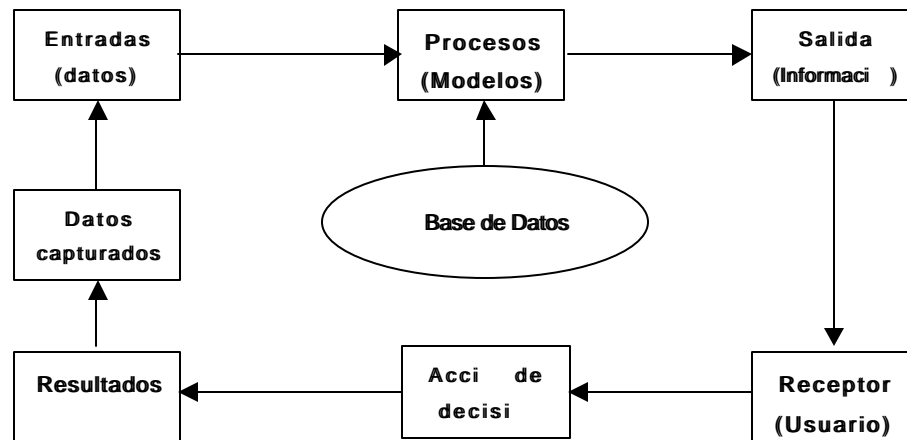


Figura 5.2. El ciclo de la información

La Figura 5.2 nos da una idea macro de las fronteras de Maxim, nos indica la forma como debemos pensar acerca del flujo de información, observe que los datos que se van a procesar pueden ser datos de entrada, estar almacenados o se pueden dar ambos casos, para después procesarlos mediante modelos para generar

información, el receptor recibe la información y luego toma una decisión, esto genera otras acciones o eventos, que a su vez crea otros datos dispersos que se capturan y sirven como entradas, y el ciclo se vuelve a repetir.

5.1.3. Identificación de los Objetivos Generales

Se han considerado los siguiente objetivos que debe de cumplir Maxim para cubrir las expectativas de análisis y cálculo de una manera eficiente, estos se los divide en dos categorías, actuariales y computacionales.

5.1.3.1. Objetivos Actuariales

Permitir el cálculo de los valores actuariales, riesgos para modelos discretos de seguros y rentas, así como primas, y reservas matemáticas, para distintas tablas de mortalidad.

Maxim debe de ser capaz de asignar a un modelo actuarial una tabla de mortalidad, donde halle los símbolos de conmutación necesarios para el cálculo de valores actuariales o riesgo.

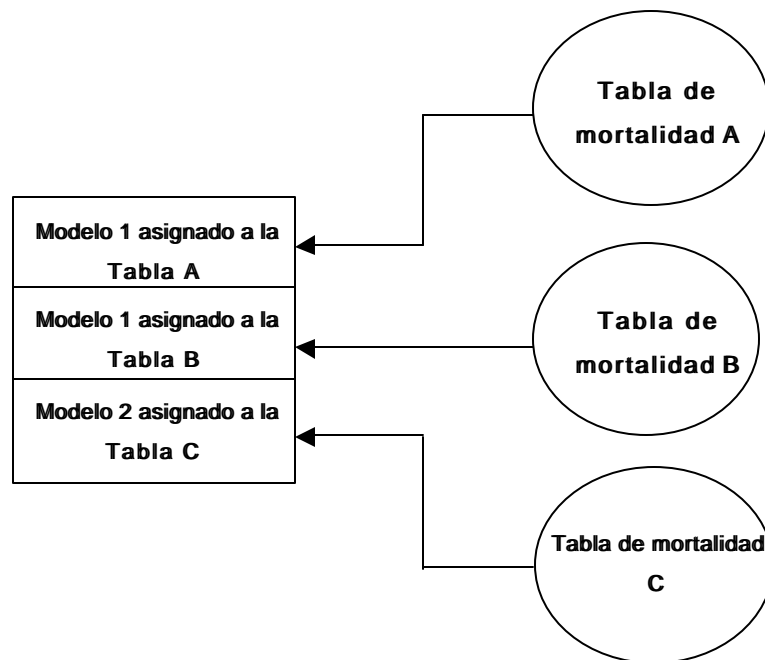


Figura 5.3. Asignación de una tabla de mortalidad a un modelo

Como se observa la tabla de mortalidad es un atributo asignado a un modelo actuarial, esto conducirá a desarrollar comparaciones entre los valores actuariales de un modelo para distintas tablas de mortalidad.

Permitir el cálculo de los valores actuariales para modelos continuos de seguros y rentas, así como primas, y reservas matemáticas, para distintas leyes de mortalidad.

Los valores actuariales donde los pagos son al momento en que acaece el evento denotado como f/q , (Sin utilizar la aproximación de los modelos discretos), requieren de una ley de mortalidad teórica, para su cálculo, por lo tanto un modelo continuo debe tener asignado una ley específica, como se observa en el grafico siguiente.

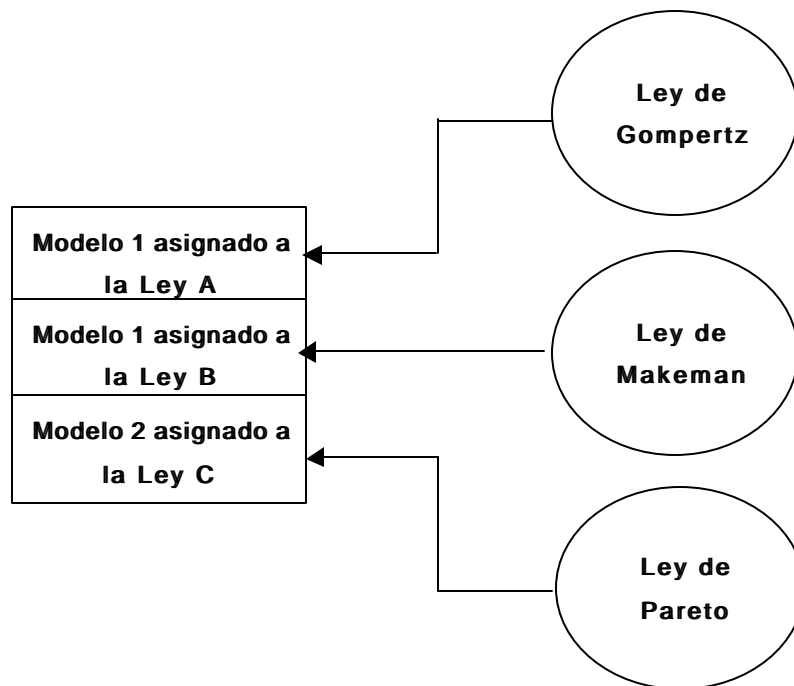


Figura 5.4. Asignación de un modelo a una ley de mortalidad

Así como un modelo discreto tiene asignado una tabla de mortalidad, un modelo continuo debe tener asignada, una ley de mortalidad y si el usuario requiere se puede comparar modelos discretos y continuos a la vez.

Permitir sensibilizar los modelos actuariales, variando los parámetros de este, para modelos de seguros o rentas de vida, continuos o discretos.

Al sensibilizar los parámetros de un modelo, se podrá analizar cual de éstos impactan más significativamente el valor actuarial y el riesgo de un modelo, estos parámetros pueden ser la tasa de interés, la edad del beneficiario, la vigencia del modelo, e inclusive la tabla o ley teórica de mortalidad entre otros parámetros.

Cálculo de valores actuariales haciendo uso de modelos continuos donde las indemnizaciones son funciones continuas.

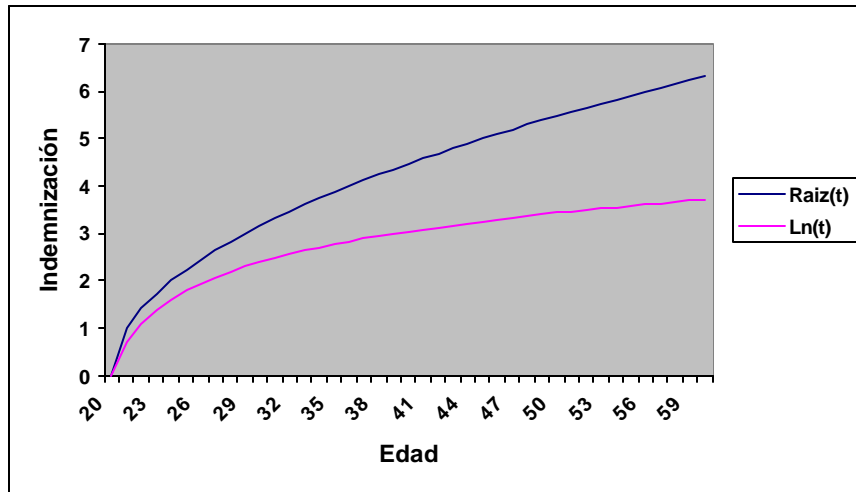


Figura 5.5. Modelos continuos con indemnización Funcional

El gráfico 5.5 nos muestra dos modelos de seguros cuya indemnización sigue un patrón de pagos en el tiempo en el caso que f/q , de manera logarítmica, y la raíz cuadrada del tiempo futuro de supervivencia, el valor actuarial de estos modelos se los puede obtener sin dificultad si se asume o se conoce la ley teórica de mortalidad de la población, de este modo se consigue tener modelos con indemnizaciones cuadráticas, exponenciales, etc.

Cálculo de valores actuariales, riesgos, primas y reservas para Planes.

Un plan se define como un conjunto de modelos (discretos o continuos) de seguros, de rentas, o combinación de ambos, ubicados dentro de un periodo de tiempo, el siguiente gráfico nos presenta la idea de un plan.

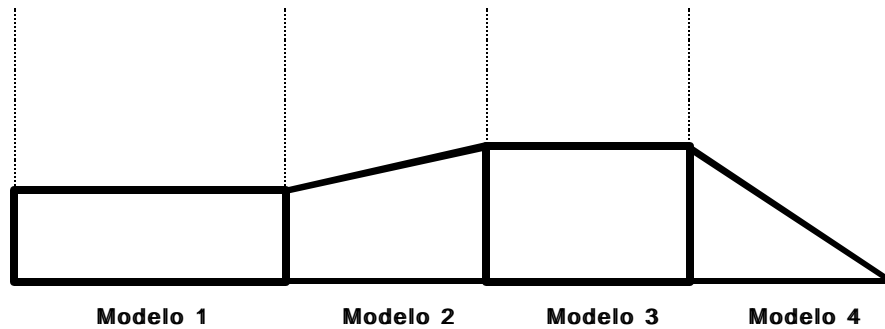


Figura 5.6. Esquema de un Plan (Plan A)

En el gráfico 5.6 se puede observar que el Plan A está formado por 4 modelos actuariales, el primero con indemnización constante temporal, el segundo un modelo incremental diferido y con vigencia, el tercero un modelo con indemnización constante diferido y temporal, y el cuarto un modelo decremental diferido, a pesar de contener distintos modelos el plan contiene un solo valor actuarial en el instante actual.

Es simple asumir considerando los objetivos actuariales que pretende cubrir Maxim, se podrá examinar varios planes a la vez, que están basados en modelos discretos asignados a distintas tablas de mortalidad, o aquellos basados en modelos continuos para distintas leyes teóricas.

5.1.3.2. Objetivos Computacionales

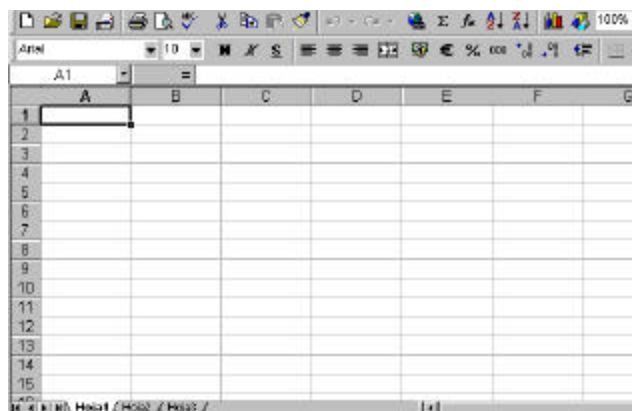


Figura 5.7. Hoja electrónica de Microsoft Excel

Las hojas electrónicas modernas como Microsoft Excel presenta una interfaz visual apropiada, para el manejo y presentación de información numérica, lo cual es parte de la esencia de Maxim, para el posterior análisis actuarial.

Las herramientas de edición presentes en todas las aplicaciones Windows, tales como copiar, pegar, cortar así como las herramientas de archivo tales como Abrir un archivo, guardar un archivo, imprimir, etc son características que Maxim debe poseer para el óptimo manejo de los datos, y por la familiaridad del usuario con estos tipos de menú.

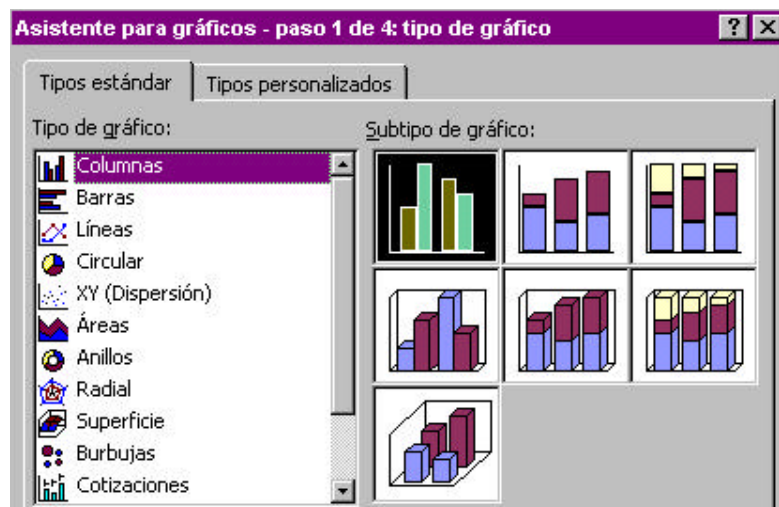


Figura 5.8. Asistente gráfico de Excel

Una herramienta básica para la interpretación y análisis de los resultados son los gráficos, estos nos dan una idea general e instantánea de la situación de los resultados, de aquí que, se debe

presentar gráficos diversos para este fin, como se observa en la figura 5.8 el asistente para gráficos de Excel.

Capacidad para manejar más de una hoja electrónica a la vez

En muchas ocasiones el usuario requerirá realizar dos o más tipos de análisis en paralelo y tenerlos organizados en hojas diferentes con su respectivo gráfico, por ejemplo supongamos que el usuario desee analizar modelos con indemnización constante asignado a una tabla de mortalidad específica, y los mismos modelos asignados a una ley teórica de mortalidad, éste puede organizar los resultados en hojas electrónicas diferentes e independientes para cada caso. Un ejemplo de este requisito nos lo da Excel, como se observa en figura 5.7, esta herramienta da la posibilidad de manejar un número de hojas limitado por la capacidad de memoria del ordenador donde este instalado la hoja electrónica, y en cada hoja existe la posibilidad de incrustar un gráfico como un objeto.

Código reutilizable capaz de ser utilizados en otras aplicaciones diferentes a Maxim, que admitan automatización Active X.

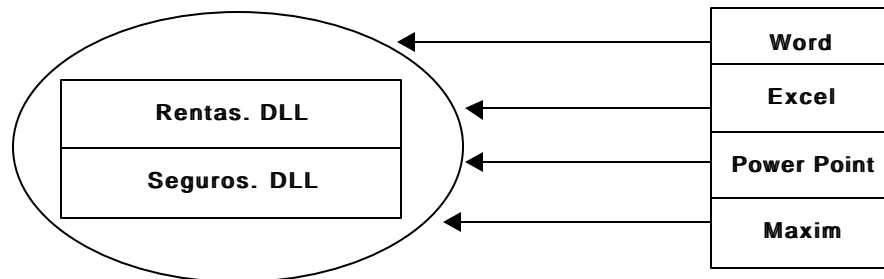


Figura 5.9. Esquema de código reutilizable

Una de las ventajas de organizar el código en componentes, es que todas las aplicaciones de Windows como Word, Excel, Power Point admiten automatización Active X, entonces Maxim al ser una aplicación que funciona en el ambiente Windows debe de soportar esta características.

Como se puede observar en el figura 5.9, las aplicaciones clientes hacen referencias a los componentes, para así utilizar las interfaces que estos componentes presentan, como sus métodos y propiedades.

5.1.4. Análisis de las necesidades del sistema

El máximo potencial de análisis de Maxim, se alcanza cuando el usuario se convierte en parte del flujo de información, como se mencionó en la definición del sistema. Pero el sistema no solo tiene como entradas los requerimientos del usuario si no también las tablas de mortalidad, y las leyes teóricas de mortalidad, mientras mayor sea el número de tablas y diversificadas (por sexo, por provincias, por profesiones, etc) Maxim acrecienta su utilidad.

Para alcanzar los objetivos planteados Maxim se divide en tres módulos dependientes entre si.

- 1) Módulo para el análisis Unimodelos
- 2) Módulo para el análisis Multimodelos
- 3) Módulo para el cálculo de los valores actuariales para modelos, continuos y discretos.

El tercer módulo es el eje principal y será la base para el buen desempeño de los dos módulos anteriores. Cada uno de los módulos tiene sus propios objetivos particulares, y su propio

alcance, pero no son independientes entre si, según los requerimientos del usuario, estos se invocan entre ellos.

El diagrama de contexto que nos da una visión macro de los procesos y las necesidades de información de Maxim se presenta a continuación.

5.1.4.1. Diagrama de contexto

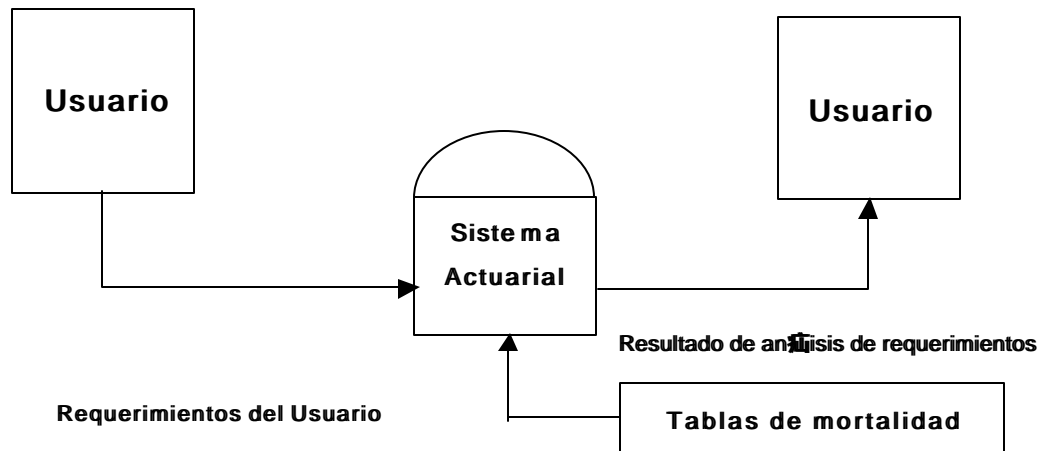


Figura 5.10. Visión General del Sistema

Como se observa tanto los requerimientos del usuario, así como los símbolos de conmutación de las tablas de mortalidad son las entradas fundamentales al proceso principal del sistema, las tablas de

mortalidad pueden estar en bases de datos diferentes, o contenidas en una misma base.

Como se señaló, esto nos da una idea general de las fronteras de Maxim, ahora como si tuviéramos una lupa de aumento procederemos a detallar un poco más, con el diagrama de nivel cero, presentado a continuación.

5.1.4.2. Diagrama cero

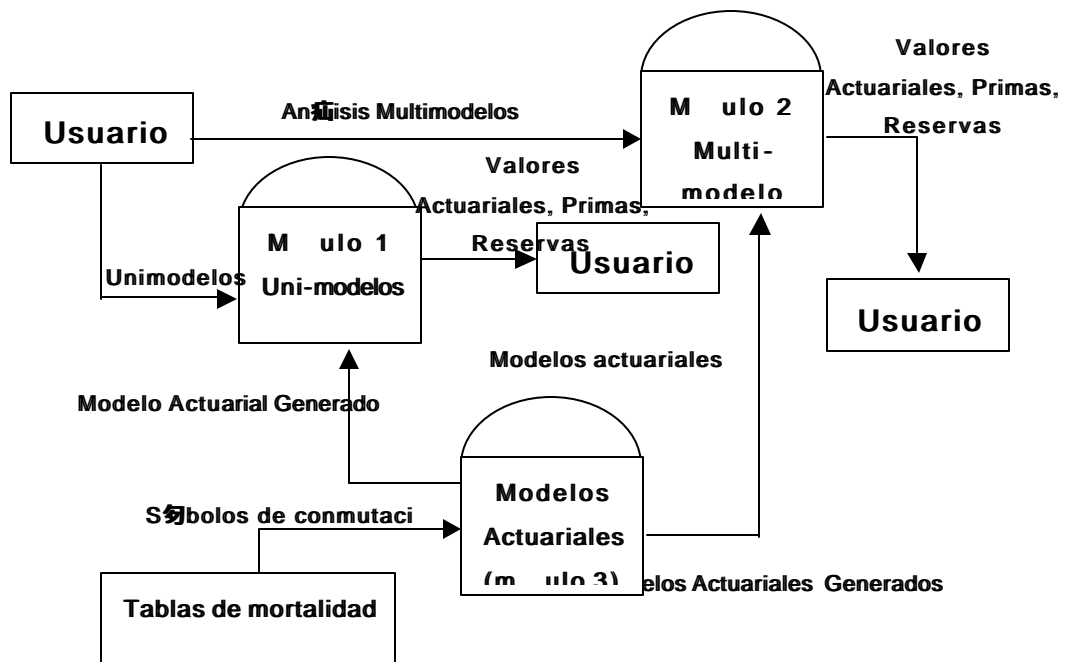


Figura 5.11. Diagrama cero

El módulo para el análisis Uni - modelo (módulo 1) pasa a estado activo cuando el usuario requiere analizar uno o más modelos actuariales de manera individual.

El módulo para el análisis Multi - modelo (módulo 2) pasa a estado activo cuando el usuario requiere analizar uno o varios planes (Vea definición de “plan” en Identificación de objetivos).

El proceso “Modelos Actuariales” (módulo3), es donde se generan los modelos tanto para rentas como seguros discretos y continuos, con las características que requiere el usuario, note que a este proceso converge la mayor parte de los flujos de datos, como se menciono este módulo es el eje de Maxim, el código que permite la generación de estos modelos serán organizado en dos componentes, DLL Active X, más adelante se detallará el diseño de estos componentes.

La información presentada al usuario será organizada y enviada por el monitor del ordenador, o por medio impreso.

Siguiendo con el detalle de los procesos, ubicamos la lupa de aumento en el “Módulo para el análisis Uni - modelo”, consecuentemente tenemos el siguiente diagrama de nivel 1.

5.1.4.3. Diagrama de Nivel 1 para el módulo de análisis Uni - modelo

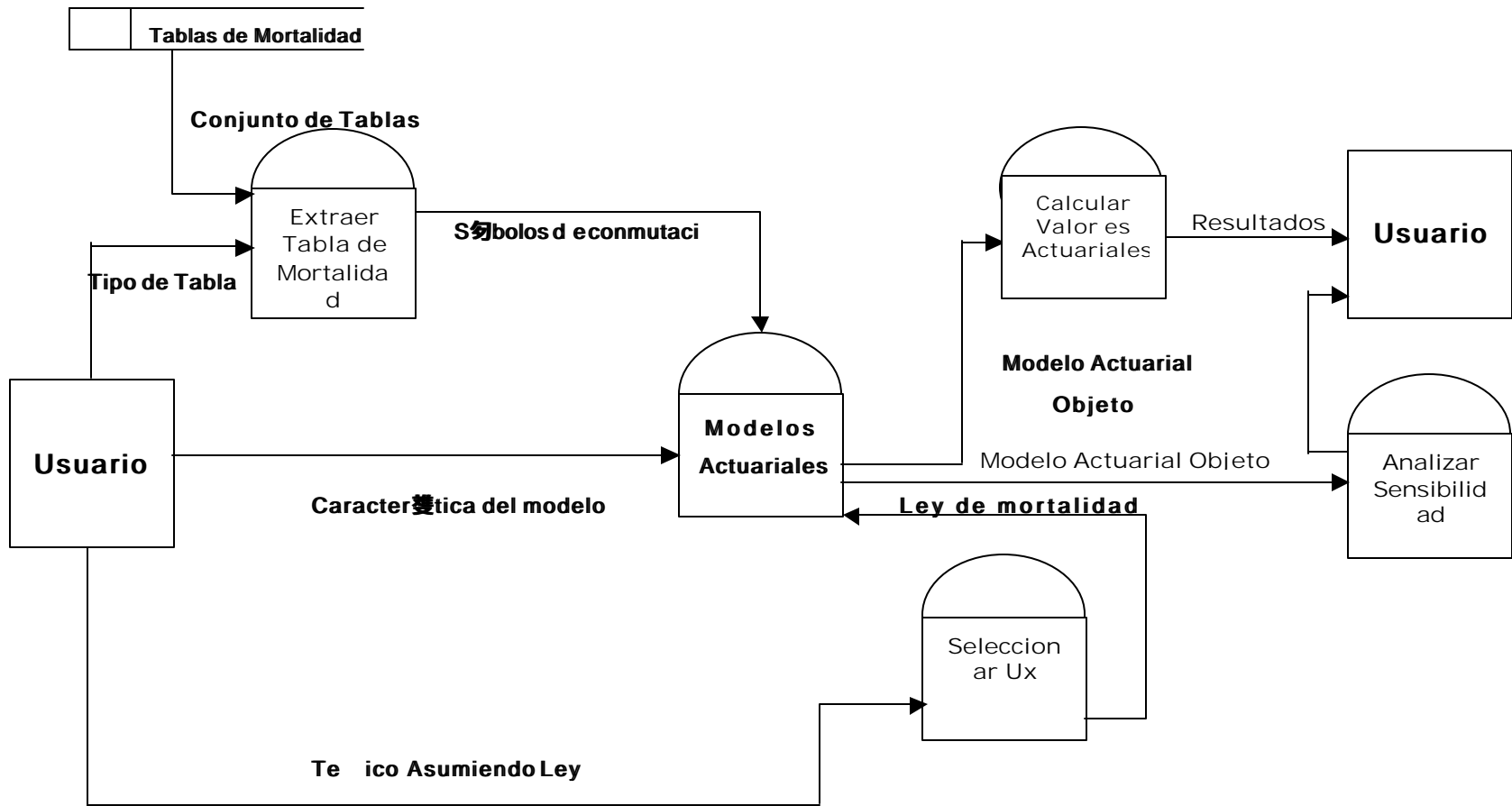


Figura 5.12. Módulo 1 para el Análisis Uní - modelos

5.1.4.3. Diagrama de Nivel 1 para el módulo de análisis Uni – modelo.

Siguiendo con el detalle de los procesos, ubicamos la lupa de aumento en el “Módulo para el análisis Uni - modelo”, consecuentemente tenemos el siguiente diagrama de nivel 1, el cual se muestra en la figura 5.12.

5.1.4.3.1. Objetivos

Analizar los modelos actuariales de seguros y rentas, de manera individual así como los distintos tipos de primas y reservas matemáticas que se pueden desarrollar con estos modelos, con enfoques de tipo discreto y continuo.

5.1.4.3.2. Alcance

El proceso permitirá al usuario el análisis comparativo de dos o más modelos teniendo como medida de comparación sus valores actuariales y riesgos, variando los parámetros de éstos como la tasa de interés, la vigencia, la edad del asegurado, el diferimiento, e inclusive las tablas o leyes teóricas de mortalidad asignados a los modelos.

Este módulo permitirá a un modelo discreto asignarle una tabla de mortalidad específica, o una ley teórica en caso de modelos continuos, con esto el usuario logrará comparar los valores actuariales de un modelo, provenientes de dos o más tablas diferentes o de leyes teóricas distintas.

5.1.4.3.3. Diccionario de datos

5.1.4.3.3.1. Proceso : Extraer Tabla de mortalidad

Salidas

Símbolos
de
conmutación

Entradas

Conjunto de tablas

Teórico Asumiendo ley

Descripción del proceso: Este proceso es el responsable de seleccionar el conjunto de tablas existentes, que requiere el usuario y asignarlas a los modelos discretos, para así extraer los símbolos de conmutación necesarios para los respectivos cálculos.

5.1.4.3.3.2. Proceso : Generar Modelos Actuariales

Salidas		Entradas	
Modelo	Actuarial	Características	del
(Objeto)		modelo	
		Requerimiento: Tabla	
		Símbolos	de
		conmutación	
		Ley de mortalidad	

Descripción del proceso: Este es uno de los procesos centrales, y es parte del módulo 3 del sistema, es el encargado de crear los modelos actuariales con las características que el usuario desea. Por ejemplo estas características pueden ser “Modelo de seguro de 1 U.M para una persona de 25 años de edad con vigencia hasta los 50 años, pagaderos al momento de f/q , donde la población sigue una ley de mortalidad de Gompertz”.

5.1.4.3.3.3. Proceso : Seleccionar Ux

Salidas		Entradas	
Ley de mortalidad		Resultado del análisis	
		Asumiendo ley de	
		mortalidad	

Descripción del proceso: Este proceso se activa cuando el usuario asume cualquier ley de mortalidad teórica, entonces el proceso asigna una tasa teórica de mortalidad a un modelo actuarial continuo.

5.1.4.3.3.4. Proceso : Calcular valores actuariales

Salidas

Valores actuariales

Entradas

Modelo actuarial
(objeto)

Descripción del proceso: Este proceso es el encargado de computar todos los valores actuariales que necesita el usuario, tales como primas netas, primas brutas, primas comerciales, riesgos, reservas matemáticas etc. Para esto necesita el o los modelos actuariales (con sus respectivas características) al que se le va a computar estos valores.

5.1.4.3.3.5. Proceso : Analizar sensibilidad

Salidas

Valores Actuariales

Entradas

Modelo actuarial
(objeto)

Descripción del proceso: Una vez que se tiene un modelo actuarial, con sus características bien definidas, se procede a analizar la sensibilidad del modelo variando sus parámetros principales, por ejemplo variamos la tasa de interés i de un modelo con un delta de 0.1% y observamos cual es el comportamiento del valor actuarial, y el riesgo de éste.

5.1.4.3.3.6. Flujo : Requerimiento de tabla

Fuente: Usuario

Destino: Extraer tabla de mortalidad

Descripción: Captura el requerimiento del usuario de trabajar sólo con tablas de mortalidad, además del nombre de la tabla, y la dirección donde se ubica la base de datos que la contiene.

5.1.4.3.3.7. Flujo : Características del modelo

Fuente: Usuario

Destino: Generar modelo actuarial

Descripción: Captura las características del modelo actuarial, con que el usuario desea trabajar, estas características pueden ser vigencia, diferido, edad del asegurado, pagos de la indemnización al momento o al final del año de finq en el caso de un modelo de seguro de vida.

5.1.4.3.3.8. Flujo : Objeto Modelo Actuarial

Fuente: Generar Objeto Modelo actuarial

Destino: Calcular valor actuarial

Descripción: Es un tipo abstracto de datos y es una instancia de una clase que representa a un modelo continuo o discreto de rentas o seguros, con las características ya definidas por el usuario.

5.1.4.3.3.9. Flujo : Conjunto de Tablas

Fuente: Base de datos

Destino: Extraer tablas

Descripción: Es un conjunto de datos abstractos que contienen una o más tablas de mortalidad, requeridas por el proceso Extraer tablas, que serán asignados a los modelos discretos respectivos.

5.1.4.4. Módulo de análisis Multi - modelo

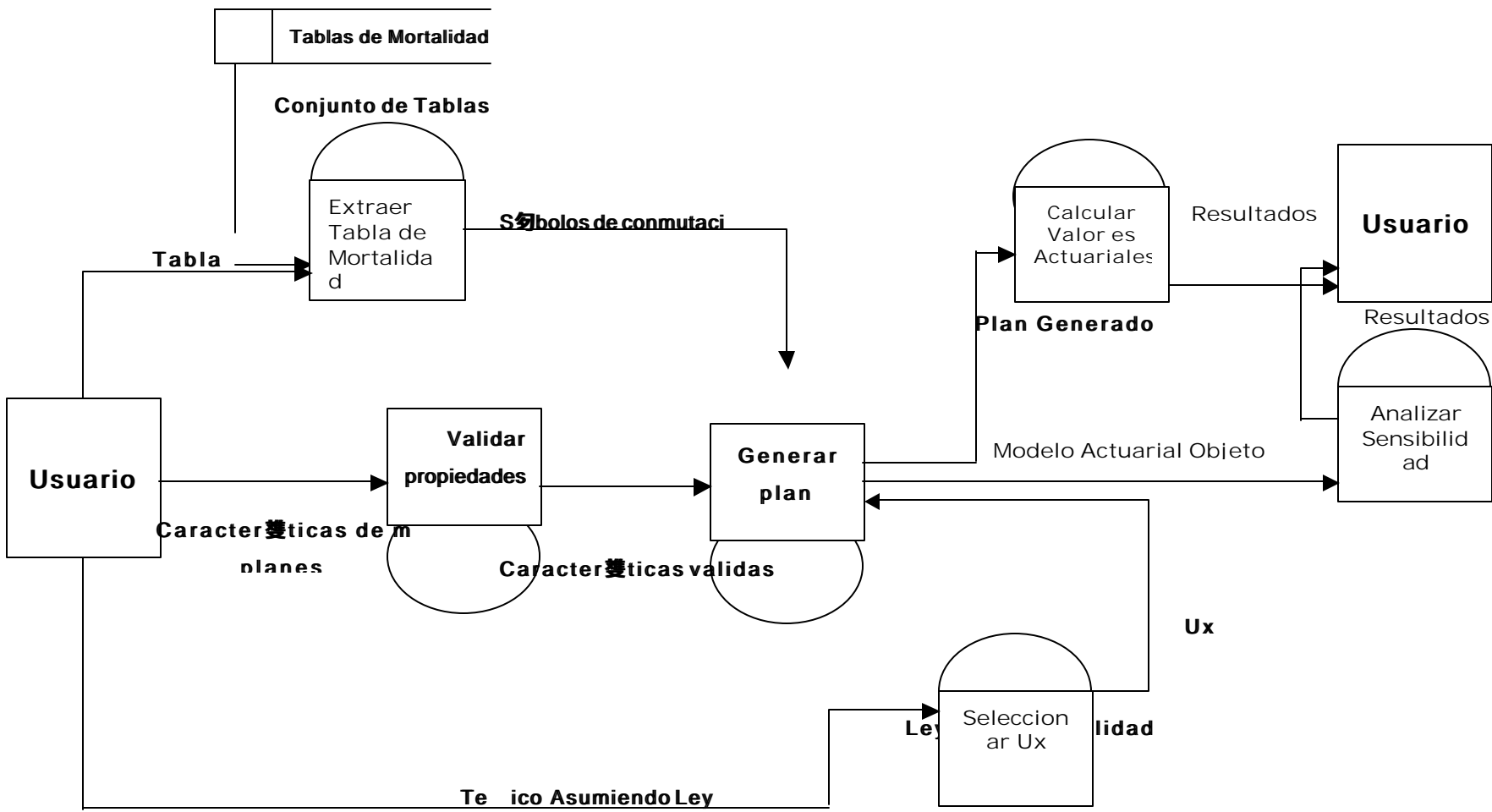


Figura 5.13. Módulo 2 para el Análisis Multi - modelos

Las compañías de seguros, que emiten pólizas de vida, o las empresas que trabajan con las rentas vitalicias, generalmente diseñan productos atractivos para atraer al cliente, y para conseguir esto combinan dos o más modelos actuariales, por ejemplo un producto diseñado para un recién nacido consiste en una renta vitalicia de 5000 dólares anuales hasta cumplir los 20 años, y un seguro de vida incremental a partir de los 20 años hasta los 30 años con una base de 5000 dólares.

Note que solo hemos combinado un modelo renta vitalicia temporal, y uno de seguro de vida diferida y temporal (esto es lo que llamamos un plan). Si la empresa que diseño este producto desea ingresarlo al mercado, la lógica indica que aparte de una investigación de mercado, un análisis actuarial es necesario para este plan. De esto es lo que se encarga el módulo 2 de Maxim.

Antes de detallar el módulo 2 es preciso definir lo que es, plan de indemnización, y plan de pagos de primas.

Plan indemnización: Es el conjunto de modelos actuariales ya sea de rentas vitalicias o seguros que beneficia al asegurado, durante un periodo n de años.

Plan de pagos: es el conjunto de modelos de rentas vitalicias, donde los pagos son las primas que el asegurado se compromete a cancelar por un Plan de indemnización.

5.1.4.4.1 Objetivos

Calcular la prima única pura, y riesgo para un plan de indemnización para un periodo de n años compuestos por un conjunto de modelo actuariales, pueden ser estos de rentas o seguros, ubicados cada uno en m particiones dentro del periodo.

Calcular la prima única pura, primas de inventario, primas comerciales, compuestos por planes de indemnización y planes de pagos de primas diseñados utilizando los modelos actuariales existentes, así como calcular las reservas matemáticas para cada una de las primas antes mencionadas.

5.1.4.4.2 Alcance

El módulo estará en capacidad de ubicar m modelos en un periodo de tiempo de n años sin permitir conflictos entre ellos, por ejemplo que las características de vigencia, y diferimiento de cada modelo sean las adecuadas.

Los valores actuariales y riesgos para los planes tendrán la flexibilidad de ser calculados para diferentes tablas de mortalidad, o para diferentes leyes teóricas.

5.1.4.4.3 Diccionario de datos

Si comparamos el diagrama 1 del análisis Uni - modelo, con el diagrama del análisis Multi - modelos se observa que ambos tienen varios procesos en común, por lo que en este diccionario de datos solo se detallarán los procesos y flujos no comunes.

5.1.4.4.3.1. Proceso : Validar Propiedades

Salidas

Características
de
validad

Entradas

Características de
los m planes

Descripción del proceso: El usuario al requerir el análisis de un plan debe de proporcionar al sistema la información necesaria de cada uno de los a parámetros de cada modelo que conforma el plan, este proceso comprueba que los parámetros ingresados sean validos y no existan dos modelos ubicados en una misma partición.

5.1.4.4.3.2. Proceso : Generar planes

Salidas

Modelos de
objetos

Entradas

Características
validas de los m
planes

Descripción del proceso: Este proceso genera todos los modelos necesarios requeridos por el usuario, tanto para los planes de indemnización, como para los planes de pagos de primas.

5.1.4.4.3.3. Flujo : Características de m planes

Fuente: Usuario

Destino: Validar características

Descripción: En este flujo contiene información correspondiente a las características de los modelos que el usuario escogió al diseñar el plan.

5.1.4.4.3.4. Flujo : Modelos actuariales generados

Fuente: Generar plan

Destino: Calcular valores actuariales

Descripción: En este flujo transita los m modelos, mas adelante veremos que cada modelo es un objeto con sus propiedades y métodos definidos, por lo tanto podemos decir que por este flujo viajan m objetos correspondientes a cada modelo con sus propiedades validadas.

5.2. Diseño

5.2.1. Diseño de las DLL Active X para modelos de seguros y rentas

Si consideramos los modelos actuariales como objetos (Intangibles), podemos identificar que estos poseen propiedades y métodos concretos, entre las propiedades tenemos las siguientes:

Tabla III

Propiedades de los modelos de seguros y rentas

Seguros		Rentas	
Propiedad	Observación	Propiedades	Observación
Tabla de mortalidad o Ley Teórica	Para Modelos discretos o Modelos continuos	Tabla de mortalidad o Ley Teórica	Para Modelos discretos o Modelos continuos
Tasa de interés	Nominal	Tasa de interés	Nominal
Diferido	Diferido a m años o Sin diferimiento	Diferido	Diferido a m años o Sin diferimiento
Indemnización	Constante Incremental Decremental Aritmético Geométrico	Pago	Constante Incremental Decremental Aritmético Geométrico
Tipo de indemnización	Al momento de f/q Al final del año f/q Fraccionario	Abono de cuotas	Anticipado Vancido

Y entre los métodos tenemos

- ✓ Método que permita calcular el valor actuarial
- ✓ Método que permita calcular el riesgo

Como se menciona en el capítulo 2, cada objeto es una instancia de una clase, que no es más que la plantilla donde se definen cada una de las propiedades, métodos y eventos que posee éste, y donde los datos se encuentran encapsulados dentro del objeto, así pues para los modelos actuariales se tendrán las clases modelos de seguros de vida, y modelos de rentas de vida, si se desea un modelo con características específicas, simplemente hay que crear una instancia de la clase respectiva con las características deseadas como se muestra en la figura siguiente.

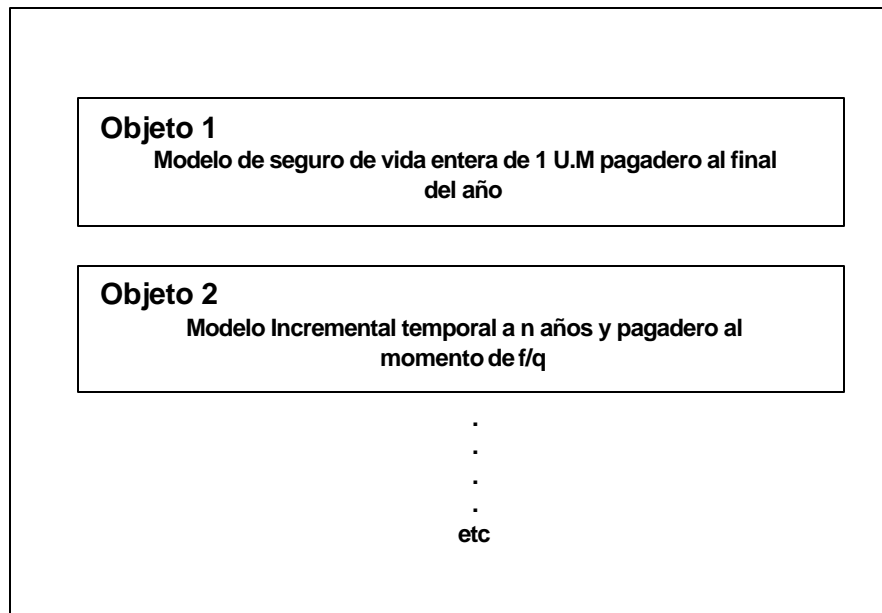


Figura 5.14. Clase Modelos de Seguros

Se diseño dos DLL Active X, Seguros.dll y Rentas.dll para que actúen como servidor para el proyecto Maxim, inclusive estos mismos componentes pueden ser utilizados por programadores, en aplicaciones clientes que Admitan automatización.

El usuario de estas componentes no debe de preocuparse de cómo estas realizan los procesos para los cálculos, pero debe de conocer las interfaces que cada componente expone, para una correcta utilización de las DLL, y por ende de los modelos que de ahí se generan.

5.2.1.1. Diseño de Seguros.dll

El principal propósito de este componente es brindar a la aplicación cliente los objetos necesarios que representen a los modelos de seguros de vida tanto continuos como discretos, para el posterior calculo del valor actuarial o riesgo.

5.2.1.1.1. Modelo de Objetos

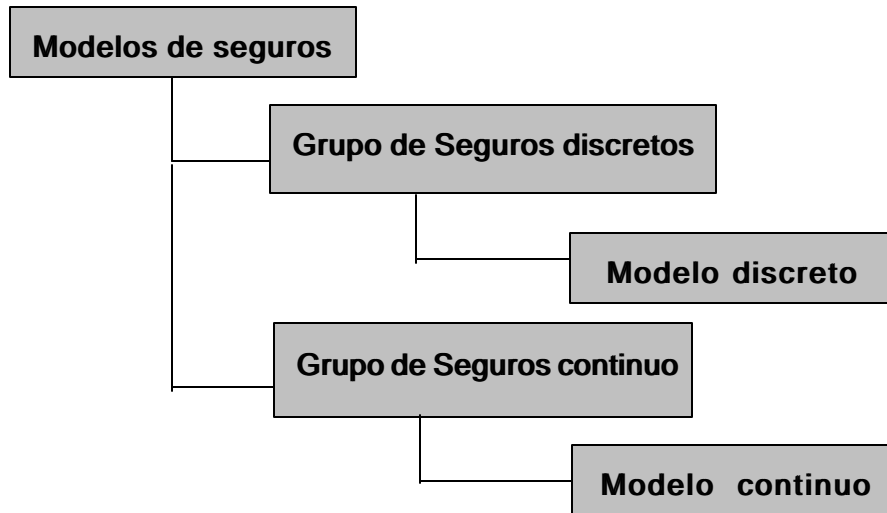


Figura 5.15. Modelo de objetos para Seguros.dll

Este componente posee las clases necesarias, para generar el modelo de objetos descrito en la figura anterior, como se observa existe una objeto contenedor "Modelos de Seguros", que enlaza tanto a los "Grupos de Seguros discretos" y "Grupos de Seguros continuo", que no son más que colecciones de objetos solamente, para Modelos discretos y continuos respectivamente, estos tienen propiedades y métodos en común pero actúan sobre distintos tipos de objetos.

5.2.1.1.2. Las colecciones de modelos continuos y discretos

Cada vez que se crea una instancia de una colección de objetos, de cualquiera de las dos colecciones mencionadas, estas presentan 3 métodos, y una propiedad que van a permitir el manejo de los modelos contenidos en la colección, los métodos son.

5.2.1.1.2.1. Método: CargarModelos

Este método carga un modelo de seguro a la colección y devuelve el modelo cargado, a su vez genera un código o identificador único de entre todos los modelos contenidos y se lo asigna al modelo que se va a añadir, en caso de un modelo de seguro discreto el identificador predeterminado es MSD - # y para un modelo de seguro continuo MSC - #, el numeral es un número entero, el formato para modelos continuos o discretos de este método es.

CargarModelo (ModeloActuarial as ModeloSeguroDiscreto)

CargarModelo (ModeloActuarial as ModeloContinuo)

5.2.1.1.2.2. Método: BorrarModelos

Este método borra un modelo de seguro contenido en una colección, identificado mediante su código.

BorrarModelo (Código de Modelo as String)

5.2.1.1.2.3. Método: ObtenerModelo

Cuando el cliente requiere un modelo de seguro específico de la colección, para su empleo, se invoca este método y se especifica el código del modelo.

ObtenerModelo (Código del Modelo as String)

5.2.1.1.2.4. Propiedad: CantModelos

Esta propiedad devuelve la cantidad de modelos de seguros contenidos en una colección, esta propiedad es de solo lectura.

CantModelos ()

Las clases más predominantes son las denominadas “Modelo de seguro discreto” y “Modelo de Seguro continuo” aquí se definen las propiedades y métodos que un modelo de seguros ya sea éste continuo o discreto debe poseer.

5.2.1.1.3. Módulo de clase: Modelo de seguro discreto

Las propiedades definidas a continuación son de lectura y escritura, que en Visual Basic se declaran con las palabras reservadas *Property let*, y *Property get*, posteriormente para referirnos a las propiedades no haremos énfasis si esta es de lectura o escritura.

5.2.1.1.3.1. Propiedad: PathBase

Esta propiedad contiene la dirección de la base de datos con extensión mdb donde se encuentra la tabla de mortalidad a la que

hará referencia el modelo, esta propiedad es una cadena de caracteres.

5.2.1.1.3.2. Propiedad: NombreTabla

Esta propiedad contiene el nombre de la tabla de mortalidad a la que hace referencia el modelo, esta propiedad es una cadena de caracteres.

5.2.1.1.3.3. Propiedad: EdadInic

Esta propiedad contiene la edad del asegurado al que se le confiere el modelo de seguro, la edad debe ser un entero.

5.2.1.1.3.4. Propiedad: Vigencia

Esta propiedad contiene la edad máxima de vigencia del modelo y ésta debe de encontrarse entre la EdadInic del modelo hasta la edad máxima que tolera la tabla de mortalidad que generalmente no es mayor a 110 años, esta propiedad es un dato entero.

5.2.1.1.3.5. Propiedad: Diferido

Si el modelo requiere la característica de diferido este debe de encontrarse entre la EdadInic y Vigencia, representa hasta que edad se difiere el modelo, si Diferido es vacío, el modelo concebirá que no existe diferimiento.

Propiedad: PeriodoDePago

Esta característica precisa si el modelo de seguros es fraccionario o no, define el periodo de tiempo que se tiene que esperar después de acaecido el evento de f/q, para recibir la indemnización pactada, este periodo puede ser un año, un semestre, un trimestre, o cualquier fracción del año. Esta propiedad toma valores enteros de 1 a 12, donde 1 es anual, y 12 es mensual.

5.2.1.1.3.6. Propiedad: MomentoDePago

Esta propiedad especifica si el pago de la indemnización es al momento en que acaece el evento de f/q, o al finalizar el Periodo de Pago.

5.2.1.1.3.7. Propiedad: TipoDeIndemnizacion

Esta propiedad define el tipo de indemnización del modelo, y estos pueden ser: Constante, Incremental, Decremental, Aritmético, Geométrico, Fraccionario.

5.2.1.1.3.8. Propiedad: Indemnización

Esta propiedad contiene la cantidad en cualquier unidad monetaria, de la indemnización a cancelar en caso de f/q, Este valor no puede ser negativo.

5.2.1.1.3.9. Propiedad: EdadMax

EdadMax contiene la edad máxima de la tabla de mortalidad a la que hace referencia el modelo generado. Esta propiedad puede ser ingresado por parte del usuario, o puede ser capturado desde la tabla de mortalidad activada por el modelo.

5.2.1.1.3.10. Propiedad: TasaTecnica

Esta propiedad recoge la tasa de interés que el modelo precisa para los cálculos respectivos, esta tasa es double y no debe ser negativa.

5.2.1.1.3.11. Propiedad: CdeltaAlfaGamma

Esta captura las constantes que requieren los modelos para la indemnización en progresión aritmética y geométrica.

5.2.1.1.3.12. Método: ErrorParametros

Para que el componente sea más robusto al momento de realizar los cálculos, este método comprueba la coherencia de las propiedades ingresadas, por ejemplo no se puede calcular el valor actuarial de un modelo si este está haciendo referencia a una tabla de mortalidad no válida. Si se encuentra algún error de ingreso o de concordancia entre propiedades, el método retorna un valor booleano FALSE a la aplicación cliente, de caso contrario retorna TRUE.

5.2.1.1.3.13. Método: MomentosSegurosDiscretos

Este método calcula el valor actuarial, y el segundo momento de la variable aleatoria que fue definida por las propiedades del modelo de ahí el nombre del método, ya que el valor actuarial en sí es el primer momento de la variable aleatoria.

En esta sección se declaran los valores actuariales de los distintos modelos, en términos de los símbolos de conmutación, la estructura principal de este método se presenta en la siguiente figura.

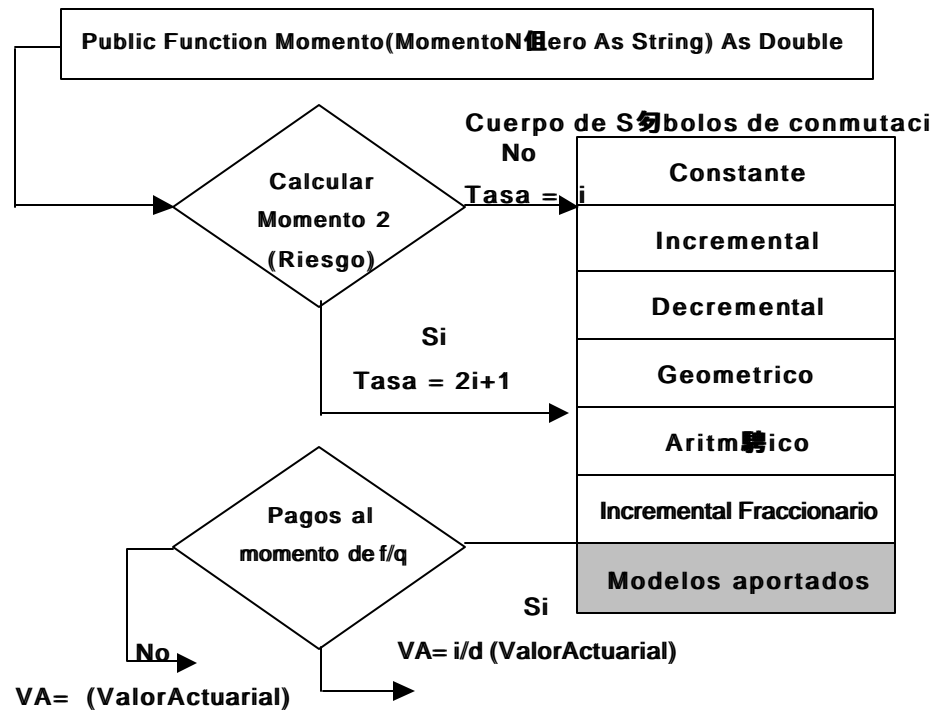


Figura 5.16. Estructura de MomentosSegurosDiscretos

Como se observa existe un cuerpo principal donde se definen los valores actuariales en términos de los símbolos de conmutación, para los distintos modelos propuestos, y al último de este cuerpo de bloques existe uno sombreado llamado Modelos aportados, esto nos indica que Seguros.dll no se limita solamente a los modelos existentes, ya definidos en el capítulo 3, si no también tenemos la posibilidad de añadir en tiempo de diseño modelos

personalizados discretos, lo que aumenta la flexibilidad de este componente.

5.2.1.1.4. Módulo de clase: modelo de seguro continuo

Este módulo genera modelos continuos enlazados con una ley de mortalidad teórica de las ya mencionadas con anterioridad, las propiedades de estos modelos son en la mayor parte igual al de los modelos discretos, pero con algunas excepciones.

5.2.1.1.4.1. Propiedad: EdadInic

EdadInic, es la propiedad que indica la edad del asegurado, esta propiedad a diferencia de su homologo en los modelos discretos es una variable double.

5.2.1.1.4.2. Propiedad: Vigencia

Vigencia, es la propiedad que indica hasta que edad rige el modelo continuo, esta variable es double.

5.2.1.1.4.3. Propiedad: Diferido

Esta propiedad indica la edad de diferimiento si existiera del modelo, si no es el caso esta edad es cero, esta variable también double.

5.2.1.1.4.4. Propiedad: Tipo de Ley

Esta propiedad indica, la ley teórica que se le asigna al modelo continuo generado, y estas pueden ser 1) Ley constante 2) Ley de Moivre 3) Ley de Gompertz, 4) Ley de Makeham, 5) Ley de Weibull, 6) Ley de Pareto. Esta propiedad toma valores de 1 a 6.

5.2.1.1.4.5. Propiedad: ConstWABCKN

Las leyes de mortalidad poseen algunas constantes, esta propiedad captura las constantes de la ley seleccionada.

5.2.1.1.4.6. Propiedad: Indemnización Funcional

El modelo actuarial al ser continuo da la posibilidad de tener funciones de indemnización continuas y convexas, esta propiedad nos da la posibilidad de escoger entre 1) Constante 2) Lineal Creciente 3) Cuadrática 3) Logaritmo natural 4) Raiz cuadrada. Esta propiedad toma valores de 1 a 4.

5.2.1.1.4.7. Propiedad: Tasa Técnica

Esta propiedad contiene la tasa técnica, necesaria para el cálculo del valor actuarial, y riesgo. Esta propiedad es double.

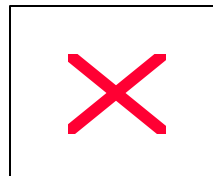
5.2.1.1.4.8. Método: Momentos Seguros Continuos

El cálculo de los momentos para seguros continuos difiere en gran medida de los discretos, puesto que las sumatorias se convierten en integrales, y los pagos no son ya al final del periodo si no al momento de f/q , y es necesario la utilización de un método numérico que permita integrar en el tiempo, el método

seleccionado por su eficacia y precisión fue Simpson 1/3, las bases de este método se detallan a continuación.

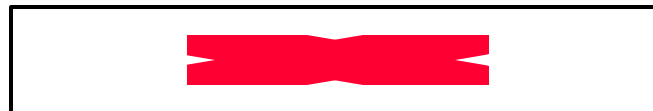
5.2.1.1.4.8.1. Bases de Método de integración

La habitual maniobra de obtener fórmulas para la integración numérica consiste en remplazar un polinomio a través de los puntos definidos por la función e integrando este polinomio obtenemos una aproximación de la integral. Así tenemos que si dentro de un intervalo seleccionamos dos puntos e interpolamos mediante un polinomio con grado $n = 1$ tenemos.



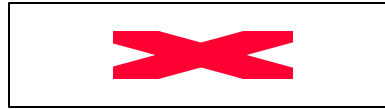
(5.1)

entonces



(5.2)

y para un intervalo de $[a - b]$ subdivido en n intervalos de tamaño igual a h tenemos.

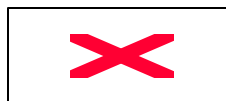


(5.3)

Esta ecuación es llamada la Regla del trapecio generalizado, es llamado así puesto que dentro de los intervalos se forman trapecios, y se toman las áreas de estos trapecios para aproximar las integrales, que son las áreas bajo las curvas.

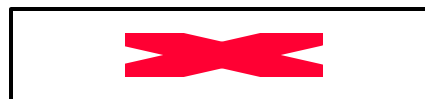
Pero este método engloba errores considerables de precisión, al menos que los intervalos sean considerablemente pequeños, esto significaría una perdida en la eficiencia del algoritmo encomendado de calcular la integral. Los errores locales (dentro de las particiones) y globales (del intervalo) se muestran a continuación.

Error Local



(5.4)

Error Total



La fórmula de integración basada en la interpolación cuadrática es conocida como la regla de Simpson 1/3, este integra un polinomio cuadrático sobre dos intervalos de igual tamaño denominado panel, considerando la formula en 5.2 tenemos la siguiente regla.

$$\int_a^b f(x) dx \approx \frac{h}{3} [f(x_0) + 4f(x_1) + f(x_2)] \quad (5.5)$$

Esta fórmula se generaliza para n sub - intervalos (panels), ubicados dentro de un intervalo [a – b], entonces tenemos.


$$\int_a^b f(x) dx \approx \frac{h}{3} [f(x_0) + 4\sum_{i=1}^{n-1} f(x_i) + f(x_n)] \quad (5.6)$$

Los errores locales y globales para este método son considerablemente menores a los anteriores, como se observa en las siguientes fórmulas.

Error Local

$$-\frac{1}{90} h^5 f^{(4)}(\xi) \quad (5.7)$$

(5.7)

Error Total 

Comparando las fórmulas 5.4 y 5.7 Simpson 1/3 presenta una precisión más efectiva. Si escogemos una interpolación cúbica o de orden superior a $n=2$, la precisión desmejora en forma significativa, debido a las oscilaciones que presenta estos tipos de polinomios, por lo que la mejor opción por precisión y eficiencia para integrar es Simpson 1/3, por lo tanto Seguros.dll utiliza este tipo de método para encontrar los valores actuariales de los modelos continuos.

5.2.1.1.4.8.2. Estructura del Método

En forma macro este método tiene la siguiente estructura:

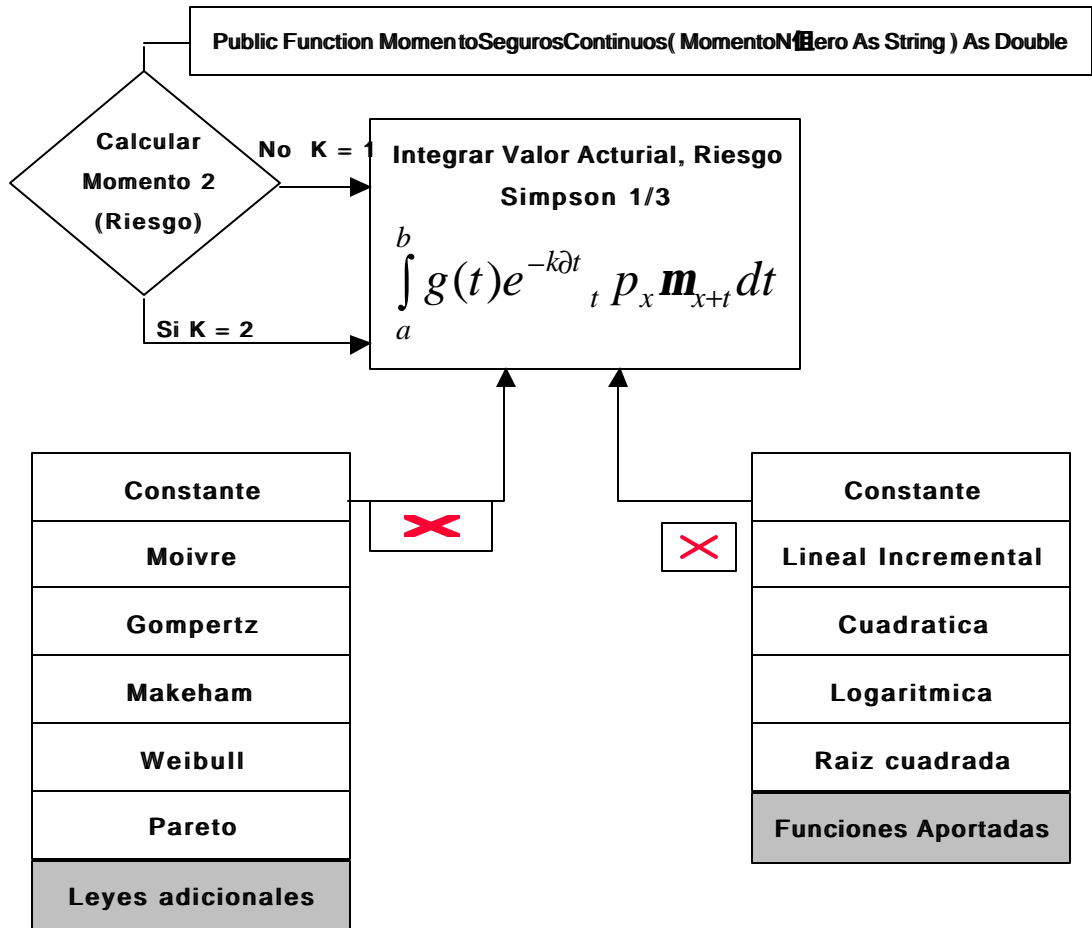


Figura 5.17. Estructura de MomentosSegurosContinuos

El proceso de integración es el proceso principal de este método como se advierte en la figura anterior. El error de integración que

se puede esperar es de aproximadamente $(0.05)^4 = 0.00000625$ puesto que el máximo h que se toma es 0.05 y el error global de Simpson 1/3 esta en el orden de $O(h^4)$.

Además se observa que existen dos bloques de funciones, una correspondiente a las leyes de mortalidad, y los otros a las indemnizaciones funcionales, se los organizó de esta manera para dar la posibilidad de que en una nueva versión de Seguros.dll el número de leyes de mortalidad e indemnizaciones aumenten sin ninguna complicación de recodificación de la componente.

5.2.1.2. Diseño de Rentas.dll

El propósito principal de éste componente es brindar a la aplicación cliente los objetos necesarios que representen a los modelos de rentas discretos de vida, para el posterior cálculo del valor actuarial y riesgo.

5.2.1.2.1 Modelo de Objetos

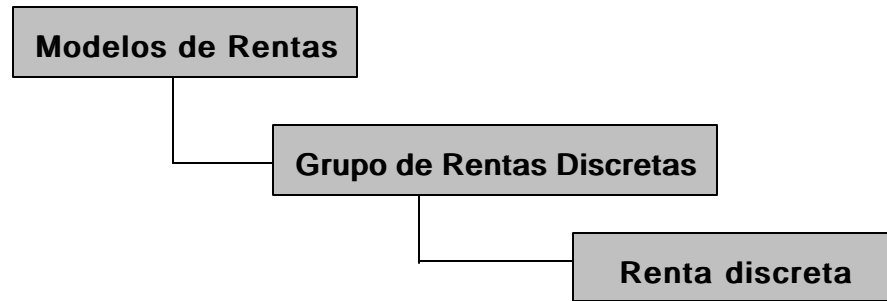


Figura 5.18. Modelo de objetos para Rentas.dll

Este modelo de objetos tiene la misma estructura que Seguros.dll para la situación de los modelos de seguros discretos, pero como ya se ha mencionado, la funcionalidad de las rentas son diferentes a las de los seguros.

Como se observa existe un objeto contenedor, teniendo como hijo la colecciones de rentas discretas esta colección es un objeto que se genera de la clases "Grupos de Rentas Discretas".

5.2.1.2.2. La colección para los modelos discretos

Cada vez que se crea una instancia de una colección, ésta presenta su interfaz constituida por 3 métodos y una propiedad, que van a permitir el manejo de los modelos de renta contenidos en la colección, estos métodos presentan la misma funcionalidad que su homólogo en Seguros.dll, claro está que las colección presentada en esta sección trabajan solo con modelos de rentas de vida discretas.

5.2.1.2.2.1. Método: CargarModelos

Este método carga un modelo de renta discreto a la colección y devuelve el modelo cargado, a su vez genera un código o identificador único de entre todos los modelos contenidos y se lo asigna al modelo que se va a añadir el código predeterminado para los modelos de rentas discretos es MRD - # donde el numeral es un número entero, el formato para modelos discretos es.

CargarModelo (ModeloActuarial as ModeloRentaDiscreta)

5.2.1.2.2.2. Método: BorrarModelos

Este método borra un modelo de renta contenido en una colección, identificado mediante su código.

BorrarModelo (Código de Modelo as String)

5.2.1.2.2.3. Método: ObtenerModelo

Cuando el cliente requiere un modelo de renta específico de la colección, para su empleo, se invoca este método y se especifica el código del modelo.

ObtenerModelo (Código del Modelo as String)

5.2.1.2.2.4. Propiedad: CantModelos

Esta propiedad devuelve la cantidad de modelos de rentas contenidos en una colección, esta propiedad es de solo lectura.

CantModelos ()

Los módulos que se van a definir continuación representan las plantillas para la generación de los objetos, que representen a las rentas discretas, y son el eje principal de este componente.

5.2.1.2.3. Módulo de clase: Modelo de Renta discreta

Las propiedades definidas a continuación son de lectura y escritura, posteriormente para referirnos a las propiedades no haremos énfasis si esta es de lectura o escritura.

5.2.1.2.3.1. Propiedad: PathBase

Esta propiedad contiene la dirección de la base de datos con extensión mdb donde se encuentra la tabla de mortalidad a la que hará referencia el modelo de renta, esta propiedad es una cadena de caracteres.

5.2.1.2.3.2. Propiedad: NombreTabla

Esta propiedad contiene el nombre de la tabla de mortalidad a la que hace referencia el modelo de renta discreto generado, esta propiedad es una cadena de caracteres.

5.2.1.2.3.3. Propiedad: EdadInic

Esta propiedad contiene la edad del tenedor de la renta vitalicia, la edad debe ser un entero.

5.2.1.2.3.4. Propiedad: Vigencia

Esta propiedad contiene la edad máxima de vigencia del modelo de renta y ésta debe de encontrarse entre la EdadInic del modelo hasta la edad máxima que tolera la tabla de mortalidad que generalmente no es mayor a 110 años, esta propiedad es un dato entero.

5.2.1.2.3.5. Propiedad: Diferido

Si el modelo de renta requiere una prórroga en el pago, éste debe de encontrarse entre las propiedades EdadInic y Vigencia, representa hasta que edad se difiere el modelo de renta, si Diferido es vacío, el modelo asumirá que no existe prórroga.

5.2.1.2.3.6. Propiedad: PeriodoDePago

Esta característica precisa si el modelo de rentas es fraccionario o no, define cual es el periodo de tiempo en que se abona una cuota al beneficiario de la renta, y esta puede ser anual, semestral, trimestral, etc. Esta propiedad toma valores discretos del 1 –12 que representan las fracciones del año.

5.2.1.2.3.7. Propiedad: TipoDePago

Esta propiedad define el tipo de Pago del modelo de renta, y estos pueden ser: Constante, Incremental, Decremental, Aritmético, Geométrico, Fraccionario.

5.2.1.2.3.8. Propiedad: Pago

Esta propiedad contiene la cantidad en cualquier unidad monetaria, del pago de la cuota en caso de que el ente sobreviva, Este valor no puede ser negativo.

5.2.1.2.3.9. Propiedad: EdadMax

EdadMax contiene la edad máxima de la tabla de mortalidad a la que hace referencia el modelo de renta generado. Esta propiedad puede ser ingresada por parte del usuario, o puede ser capturado desde la tabla de mortalidad activada por el modelo.

5.2.1.2.3.10. Propiedad: TasaTecnica

Esta propiedad recoge la tasa de interés que el modelo precisa para los cálculos respectivos, esta tasa es double y no debe ser negativa.

5.2.1.2.3.11. Propiedad: CdeltaAlfaGamma

Esta captura las constantes que requieren los modelos para los pagos en progresión aritmética y geométrica.

5.2.1.2.3.12. Propiedad: AbonoDeCuota

Esta propiedad indica si los abonos de los Pagos son 1) anticipados o 2) vencidos, por lo que esta propiedad toma solo los valores enteros 1 o 2.

5.2.1.2.3.13. Propiedad: CompErrores

Esta componente también posee un método de comprobación de errores, con lo cual garantiza al método una performance óptima a “MomentosRentasDiscretas” definido a continuación.

5.2.1.2.3.14. Método: MomentosRentasDiscretas

Este método calcula el valor actuarial, y el segundo momento de la variable aleatoria que fue definida por las propiedades del modelo de renta generado.

En esta sección se declaran los valores actuariales de los distintos modelos, en términos de los símbolos de conmutación, la estructura principal de este método se presenta en la siguiente figura.

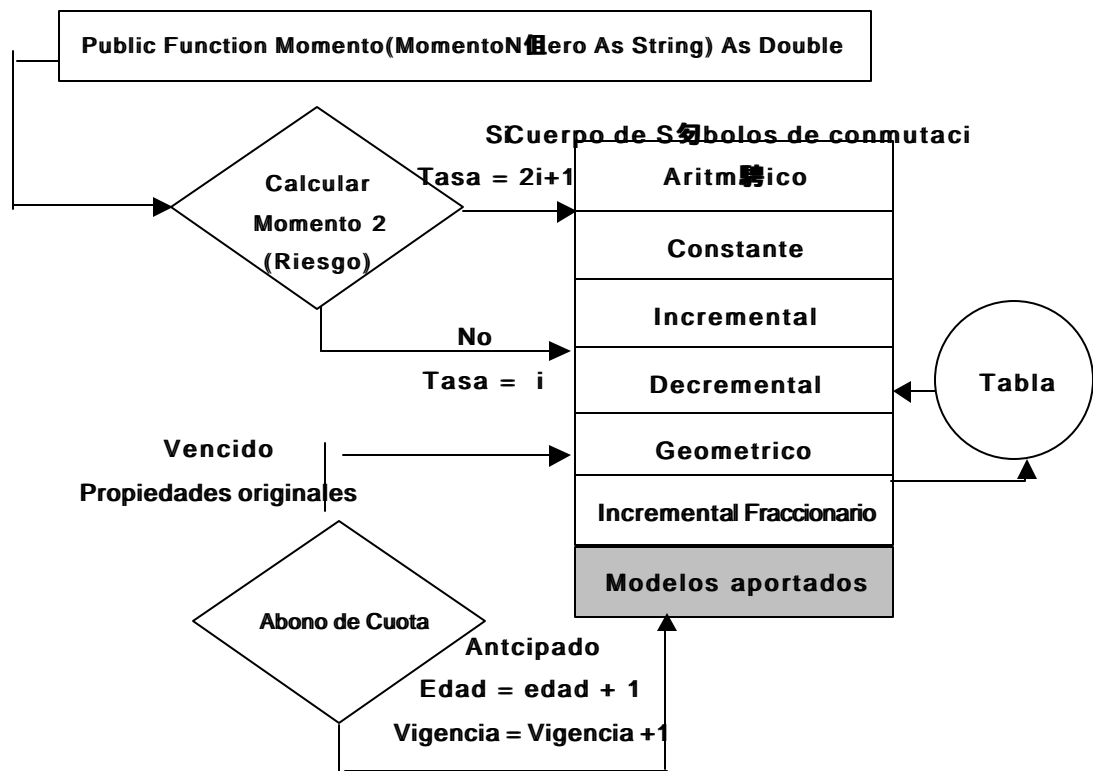


Figura 5.19. Estructura de MomentosRentasDiscretas

El cuerpo de símbolos de conmutación es donde converge todo el flujo de información que fue capturada con las propiedades del modelo, se observa que los símbolos de conmutación se han organizado de acuerdo al tipo de pago, esto nos da la posibilidad de ampliar la gama de modelos de renta con pagos discretos personalizados, sin problemas en la recodificación de la componente. De la misma manera ocurre para la situación del cálculo para el segundo momento, el método asigna la tasa respectiva para este fin.

Para los modelos con abono de cuota o pagos anticipados, simplemente se aumenta en uno las edades iniciales, de vigencia y de diferimiento si es el caso, con la seguridad que el cuerpo de símbolos de conmutación sabrá como tratar esta situación.

5.2.2. Diseño de la interfaz visual

5.2.2.1. El Modelo de Objetos

Para el manejo eficiente de los objetos, hoja electrónica y gráfico de Maxim se considera el siguiente modelo de objeto.

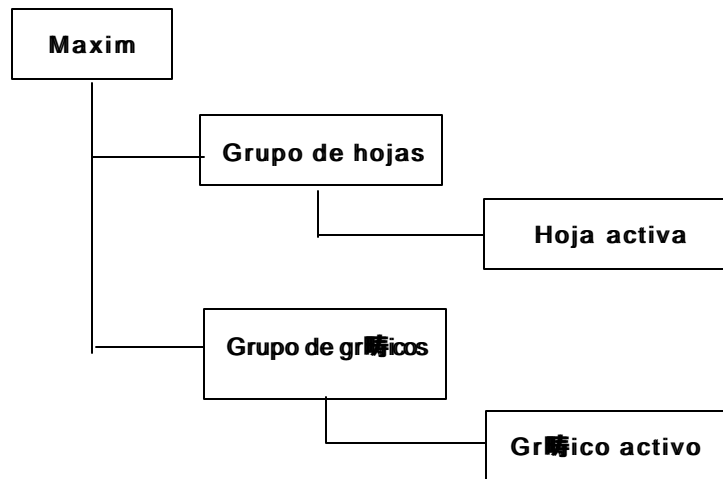


Figura 5.20. Modelo de objetos de la interfaz gráfica de Maxim

La colección grupo de hojas contiene a todas las hojas creadas por el usuario y que están siendo utilizadas por este, la hoja electrónica activa es aquella que forma parte de la interfaz actual de Maxim incluida también dentro de la colección de hojas.

Cada hoja electrónica puede o no tener un gráfico, de aquí que la colección de Grupo de gráficos posee todos los objetos gráficos pertenecientes a aquellas hojas que tengan uno.

5.2.2.1.1. La colección grupo de hojas

5.2.2.1.1.1. Método : AñadirHoja

Este método añade una hoja electrónica a la colección, identificada por un código cuyo formato es HojaMaxim# donde el numeral representa el orden dentro de la colección.

AñadirHoja(HojaActual As Hoja, CodHoja As String)

5.2.2.1.1.2. Método : RemoverHoja

Este método remueve una hoja dentro de la colección, identificado por su código, a su vez devuelve la hoja a removida.

RemoverHoja(Cod As String) as hoja

5.2.2.1.1.3. Método : ObtenerHoja

Cuando el usuario requiera cualquier hoja dentro de la colección para su operación, invoca este método y obtiene la hoja requerida identificado por su código.

ObtenerHoja(Cod As String) As Hoja

5.2.2.1.1.4. Propiedad : NumeroDeHojas

Esta propiedad devuelve el número de hojas electrónicas contenidas dentro de la colección, esta propiedad es de solo lectura, es decir no es posible asignarle un valor externo al objeto.

NumeroDeHojas() As Integer

5.2.2.1.2. Módulo de Clase para el objeto Hoja electrónica

El objeto Hoja electrónica posee una interfaz visual (que se presentará posteriormente en este capítulo) conformada por dos componentes ya existentes en VB, un contenedor y la hoja electrónica, pero para este proyecto se procedió a personalizar varios métodos y propiedades acordes con los objetivos de Maxim y se los organizó dentro de un módulo de clase denominado "Hoja".

5.2.2.1.2.1. Propiedad : Nombre

Esta propiedad guarda el nombre la hoja electrónica, es de lectura y escritura y está definido como una cadena de caracteres.

5.2.2.1.2.2. Propiedad : Activa

Esta propiedad es de lectura y escritura, e indica si la hoja es una hoja activa (es decir si está siendo utilizada actualmente por el usuario) o inactiva está definida como boolean.

5.2.2.1.2.3. Propiedad : Ventana

Esta propiedad indica la ubicación visual de la ventana esta puede ser 1) Ventana completa 2) Mosaico horizontal 3) Mosaico vertical, es una propiedad de lectura y escritura.

5.2.2.1.2.4. Propiedad :Clave

Esta propiedad contiene la clave de una hoja electrónica definida como HojaMaxim, es una propiedad de lectura y escritura.

5.2.2.1.2.5. Propiedad : Negrita

Indica si una celda o bloque de celdas contiene o no sombreado de negrita, es una propiedad de lectura y escritura.

5.2.2.1.2.6. Propiedad : Subrayado

Indica si una celda o bloque de celdas contiene o no la propiedad subrayado de fuente, es una propiedad de lectura y escritura.

5.2.2.1.2.7. Propiedad: Cursiva

Indica si una celda o bloque de celdas contiene o no la propiedad Cursiva para la fuente, es una propiedad de lectura y escritura.

5.2.2.1.2.8. Propiedad: Nuevo

Indica si una hoja electrónica es recién creada, o es una hoja abierta ya existente en el disco, es una propiedad de lectura y escritura.

5.2.2.1.2.9. Propiedad: HojaPath

Esta propiedad guarda la dirección de almacenamiento (path) desde donde fue abierta, en el caso de abrir una hoja ya existente, es una propiedad de lectura y escritura.

5.2.2.1.2.10. Propiedad: NombreFuente

Indica el nombre de la fuente establecida en una celda activa, es un propiedad de lectura y escritura.

5.2.2.1.2.11. Propiedad: RangoSeleccionado

Devuelve el rango seleccionado activo, es una propiedad de solo lectura.

5.2.2.1.2.12. Propiedad: GraficoAdjunto

Esta propiedad indica si una hoja posee un gráfico o no, es una propiedad de lectura y escritura.

5.2.2.1.2.13. Propiedad :Celda

Devuelve o establece los valores de la celda activa, es una propiedad de lectura y escritura.

5.2.2.1.2.14. Propiedad :CadenaEnBloqueSelect

Esta propiedad devuelve FALSE si entre las celdas de un bloque seleccionado activo existe alguna que contenga cadena de caracteres, de caso contrario devuelve TRUE.

5.2.2.1.2.15. Propiedad: FilaActiva

Esta propiedad devuelve la fila activa, es de solo lectura.

5.2.2.1.2.16. Propiedad: ColActiva

Esta propiedad devuelve la columna activa, es de solo lectura.

5.2.2.1.2.17. Método : Copiar

Este método copia los elementos contenidos de un bloque seleccionado.

5.2.2.1.2.18. Método :Pegar

Este método Pega en un bloque o celda los elementos contenidos en el portapapeles de Windows.

5.2.2.1.2.19. Método :Cortar

Este método Corta los elementos contenidos en un bloque o celda seleccionada al portapapeles de Windows.

5.2.2.1.2.20. Método :Imprimir

Este método ejecuta la acción de imprimir una hoja electrónica con la configuración personalizada.

5.2.2.1.2.21. Método: ImpConfigurarMargenesSuperior

Permite antes de imprimir configurar el tamaño del margen superior de la hoja de papel hasta el borde superior de la hoja electrónica.

5.2.2.1.2.22. Método: ImpConfigurarMargenIzquierdo

Permite antes de imprimir configurar el tamaño del margen izquierdo de la hoja de papel hasta el borde izquierdo de la hoja electrónica.

5.2.2.1.2.23. Método : ImpConfigurarMargenInferior

Permite antes de imprimir configurar el tamaño del margen inferior de la hoja de papel hasta el borde inferior de la hoja electrónica.

5.2.2.1.2.24. Método : ImpConfigurarMargenDerecho

Permite antes de imprimir configurar el tamaño del margen derecho de la hoja de papel hasta el borde derecho de la hoja electrónica.

5.2.2.1.2.25. Método : ImpEncabPiePagina

Permite establecer un encabezado y pie de página a la hoja de papel que se va a imprimir.

5.2.2.1.2.26. Método: OrientacionDeHoja

Orienta la hoja electrónica en sentido vertical u horizontal según los requerimientos del cliente

5.2.2.1.2.27. Método: ImpEncabCo

Este método nos da la posibilidad de imprimir la cabecera de las columnas de la hoja electrónica.

5.2.2.1.2.28. Método ImpEncabRow

Este método nos da la posibilidad de imprimir la cabecera de las filas de la hoja electrónica.

5.2.2.1.2.29. Método : ImpLineasHojas

Este método brinda la oportunidad de imprimir las líneas de división de una hoja electrónica activa.

5.2.2.1.2.30. Método :Mostrar

Este método hace visible o invisible una hoja electrónica

5.2.2.1.2.31. Método : Guardar

Este método guarda una hoja electrónica, si la hoja fue abierta con anterioridad sólo se guardarán los cambios realizados.

5.2.2.1.2.32. Método: ColorBloque

Este método asigna un color al bloque activo seleccionado

5.2.2.1.2.33. Método: NegritaBloque

Este método sombrea a todos los elementos de un bloque seleccionado, o le quita el sombreado según sea el estado de sombreado del bloque.

5.2.2.1.2.34. Método: CursivaBloque

Este método convierte todos los caracteres de un bloque seleccionado a cursiva o le inhibe esta propiedad según sea el estado de cursiva del bloque.

5.2.2.1.2.35. Método: FuenteBloque

Este método asigna a un bloque seleccionado un tipo de fuente, seleccionada de aquellas que se encuentran predeterminadas en el computador.

5.2.2.1.2.36. Método: TamañoFuenteBloque

Este método Asigna el tamaño de la fuente a los elementos de un bloque seleccionado.

5.2.2.1.2.37. Método: AlineaCentrado

Este método alinea en el centro de las celdas a todos los elementos contenidos en un bloque seleccionado

5.2.2.1.2.38. Método: Alinealzquierda

Este método alinea a lado izquierdo de las celdas todos los elementos contenidos en un bloque seleccionado

5.2.2.1.2.39. Método: AlineaDerecha

Este método alinea a lado derecho de las celdas todos los elementos contenidos en un bloque seleccionado

5.2.2.1.2.40. Método: AñadirEliminarCol

Este método elimina la columna seleccionada de una hoja electrónica

5.2.2.1.2.41. Método: AñadirEliminarFila

Este método elimina la fila seleccionada de una hoja electrónica

5.2.2.1.3. La colección Gráficos

5.2.2.1.3.1. Método : AñadirGráfico

Este método añade un gráfico a la colección de gráficos identificados por su código, cuyo formato es Graf# donde el numeral indica el orden del gráfico dentro de la colección.

5.2.2.1.3.2. Método : RemoverGráfico

Este método remueve un gráfico de la colección identificado por su código y da como resultado el gráfico removido.

5.2.2.1.3.3. Método: ObtenerGráfico

Este método devuelve el objeto gráfico identificado por su código.

5.2.2.1.3.4. Propiedad: NúmeroDeGráficos

Esta propiedad nos indica el número de gráficos existentes en una colección.

5.2.2.1.4. Módulo de clase para el objeto gráfico

El objeto gráfico al igual que la hoja electrónica posee una interfaz gráfica visible, debido a que también está compuesto por dos objetos existentes en VB. Un objeto contenedor y el objeto gráfico en si, al igual que la hoja electrónica se ha personalizado varias propiedades y este código se lo organizó en un módulo de clase denominado "Gráfico".

5.2.2.1.4.1. Propiedad : NombreGrafico

Esta propiedad indica el nombre del gráfico, es una propiedad de lectura y escritura.

5.2.2.1.4.2. Propiedad: Ventana

Esta propiedad contiene la ubicación y posición del gráfico, estos pueden ser 1)Ventana completa 2)Mosaico Horizontal 3)Mosaico Vertical, es una propiedad de lectura y escritura.

5.2.2.1.4.3. Propiedad: NúmeroDeSeries

Esta propiedad contiene el número de series (Columnas) que un gráfico va a poseer, es una propiedad de lectura y escritura.

5.2.2.1.4.4. Propiedad :TipoDeGráfico

Esta propiedad contiene el tipo de gráfico, de los cuales se tiene 1) Gráfico de barras 2) Gráfico de series de tiempo 3) Gráfico de pastel, está propiedad es de lectura y escritura.

5.2.2.1.4.5. Propiedad: GraficoVisible

Esta propiedad indica si el gráfico es visible o no, es una propiedad de lectura y escritura.

5.2.2.1.4.6. Propiedad : NombreSeries

Esta propiedad asigna o devuelve el nombre de una serie, esta es propiedad de lectura y escritura.

5.2.2.1.4.7. Propiedad: ConfigurarGráfico

Esta propiedad es de sólo lectura, y configura el gráfico considerando el tipo de monitor y la tarjeta de video.

5.2.2.1.4.8. Propiedad: NotaEjeX

Esta propiedad asigna o devuelve la leyenda en el eje de las X, es una propiedad de lectura y escritura.

5.2.2.1.4.9. Propiedad NotaEjeY

Esta propiedad asigna o devuelve la leyenda en el eje de las Y, es una propiedad de lectura y escritura.

5.2.2.2. Diseño de la Interfaz Visual

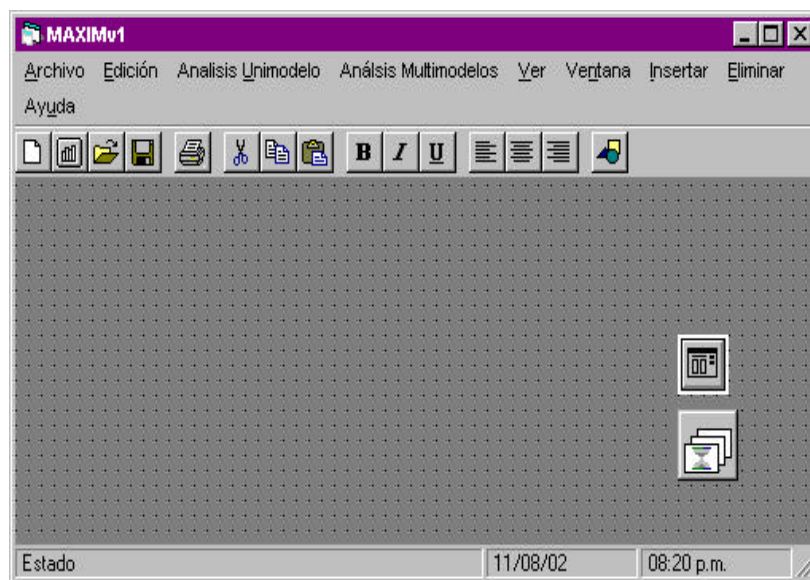


Figura 5.21. Ventana Principal de Maxim

La ventana principal de Maxim esta representada por un formulario MDI (Interfaz de múltiples documentos) el cual es contenedor de todas las hojas electrónicas y gráficos con las que el usuario está trabajando, esta ventana se encuentra constituida por la barra de menú, de donde se puede acceder a la interfaz de los dos módulos principales de Maxim, así como a los menús de edición, manipulación de archivos y formato.

Otro componente es la barra de herramientas que está conformada por las acciones más comunes que realiza el usuario dentro del sistema, entre los cuales se tiene abrir o guardar una hoja, crear una nueva hoja o un nuevo gráfico, las herramientas de edición cortar copiar y pegar, por ultimo las de formato como alinear el texto de un bloque de celdas.

5.2.2.2.1. Menú archivo

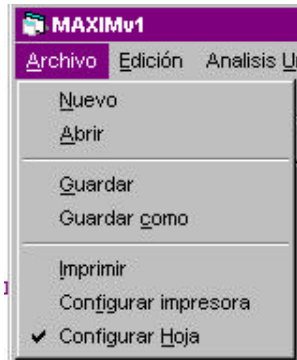


Figura 5.22. Menú archivo

El menú archivo contiene todas las acciones encargadas de la manipulación de la hoja electrónica (cuya extensión es SPP) como Crear una nueva hoja, abrir una nueva hoja, guardar una hoja abierta o nueva, imprimir la hoja activa y configurara la hoja para la impresión.

5.2.2.2 Menú Edición

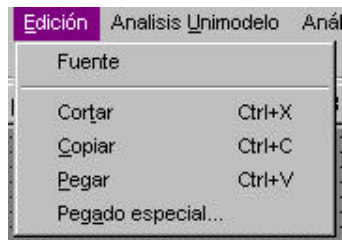


Figura 5.23. Menú Edición

Este contiene las acciones que permiten editar la información contenida en una hoja electrónica, como cortar, pegar, copiar y definir el tamaño o el estilo de la fuente, estas acciones se encuentran organizadas en procedimientos que utilizan las API de Windows.

5.2.2.2.3. Menú Análisis unimodelo



Figura 5.24. Menú Análisis unimodelo

Este menú permite acceder a la interfaz visual del modulo 1 de Maxim Análisis unimodelo, este nos brinda las opciones de generar modelos actuariales de seguros o rentas y a su vez sensibilizar estos modelos generados.

5.2.2.2.4. Menú Análisis Multiimodelo

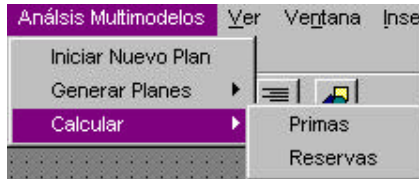
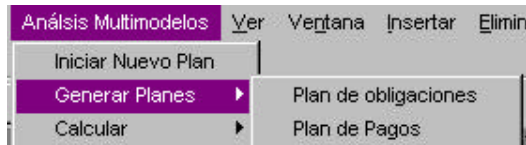


Figura 5.25. Menú Análisis Multiimodelo

Este módulo permite tener acceso a la interfaz visual del módulo 2 de Maxim Análisis multi modelos, este presenta las opciones de iniciar un nuevo plan para después generar los planes de obligaciones y pagos tendiendo como base los modelos actuariales existentes y culminar con el cálculo de las primas y reservas matemáticas de estas.

5.2.2.2.5. Menú Ver

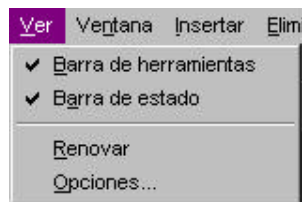


Figura 5.26. Menú Ver

Este menú nos da la opción de ocultar la barra de herramientas o de estado si así es el requerimiento por el usuario.

5.2.2.2.6. Menú Ventana

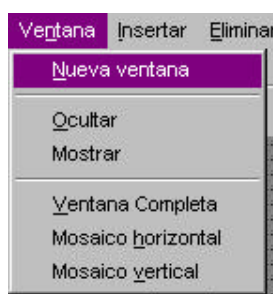


Figura 5.27. Menú Ventana

Este presenta las opciones para ocultar o mostrar una ventana o gráfico activo, así como organizarlos como mosaicos horizontales, verticales o como ventanas completas.

5.2.2.2.7. Menú Insertar

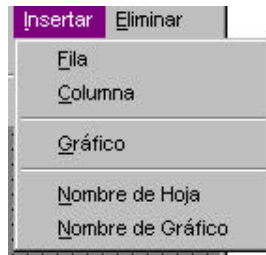


Figura 5.28. Menú Insertar

Este brinda la posibilidad de eliminar una fila o columna a la hoja activa o eliminar un gráfico asignado a una hoja electrónica.

5.2.2.2.8. Menú Eliminar



Figura 5.29. Menú Eliminar

Este brinda la posibilidad de insertar una fila o columna a la hoja activa o asignarle un gráfico.

5.2.2.2.9. Menú Ayuda

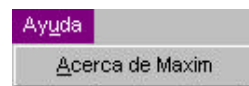


Figura 5.30. Menú Ayuda

Este menú brinda información básica acerca la hoja electrónica actuarial Maxim, como la versión o los derechos.

5.2.2.2.10. El administrador de hojas y gráficos

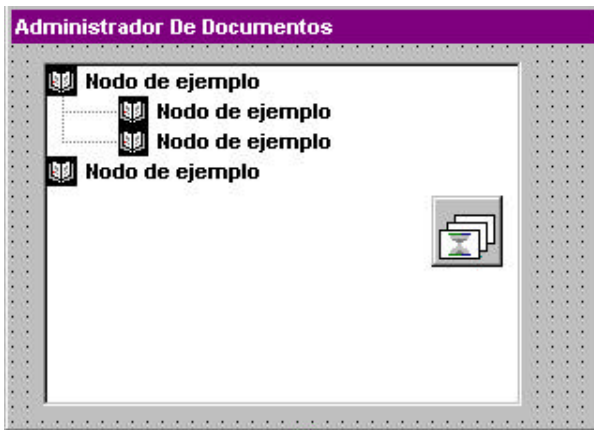


Figura 5.31. Administrador de hojas y gráficos

El administrador de hojas y gráfico permite conocer de manera visual cuantas hojas se encuentran abiertas y cuantas tienen gráficos adjuntos. También nos permite trasladarnos entre hojas sin dificultad y ver el contenido de cada una de ellas.

Para organizar las hojas y gráficos se utilizó el componente estándar de visual Basic TreeView los nodos principales son el nodo documento, hoja, y gráfico, el nodo documento contiene a una hoja electrónica con su respectivo gráfico.

5.2.2.2.11. El formulario de la hoja electrónica

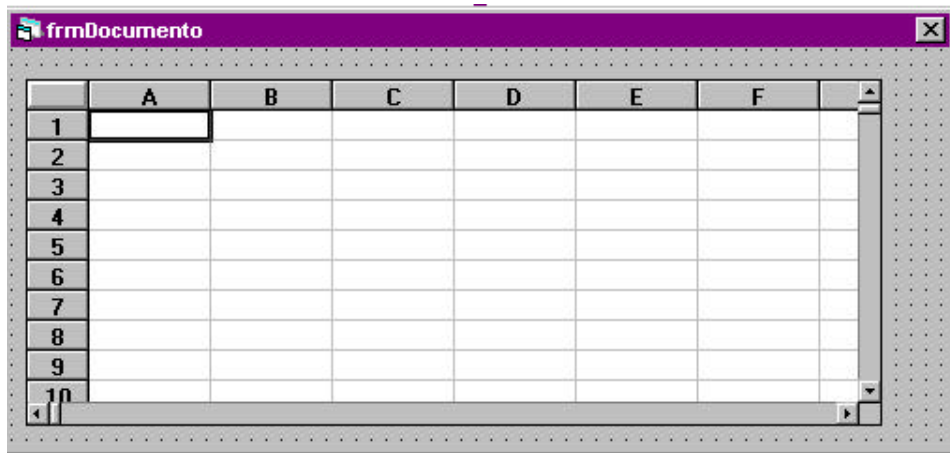


Figura 5.32. Formulario hoja electrónica Maxim

Este formulario es un contenedor de la hoja de cálculo y estos dos componentes tanto la hoja de cálculo como el formulario contenedor representan la hoja electrónica de Maxim. Además representa la interfaz visual del objeto hoja descrito anteriormente.

Interiormente Maxim maneja la hoja activa en una variable de objeto pública denominada `mpHojaActiva`, así cada vez que el usuario requiera cualquier hoja creada y contenida en la colección de hojas, esta es signada a `mpHojaActiva` y se convierte en la nueva hoja activa de Maxim.

5.2.2.2.12. El formulario del gráfico

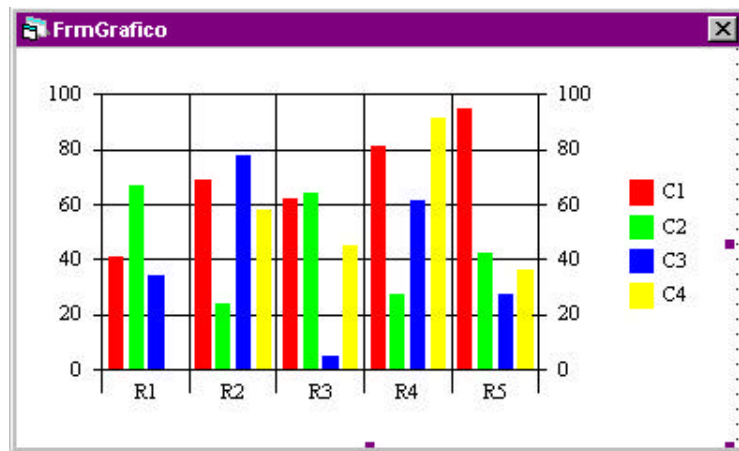


Figura 5.33. Formulario gráfico Maxim

Al igual que la hoja electrónica, la interfaz del objeto gráfico posee dos componentes un contenedor y el componente gráfico estándar de visual Basic denominado Chart, como se observa en la Fig 5.33.

5.2.2.2.13. El formulario Configurar Hoja

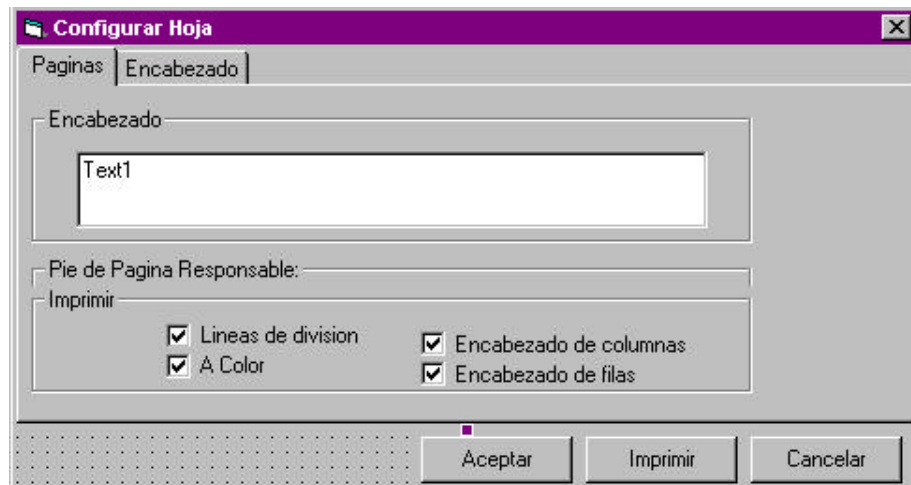


Figura 5.34. Configurar hoja

Esta forma permite configurar una hoja electrónica antes de imprimirla, nos da la opción de ubicar una leyenda en el pie de página así como imprimir las cuadrículas o los encabezados de las columnas y filas.

5.2.2.2.14. El formulario para el asistente de gráficos

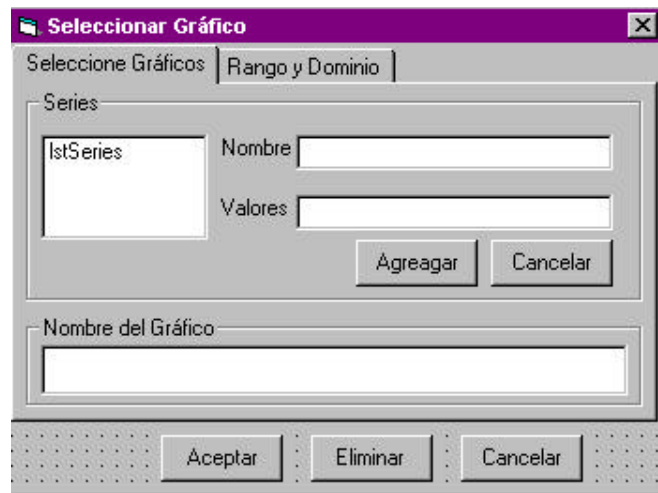


Figura 5.35. Formulario para el asistente gráfico

Maxim brinda la posibilidad de configurar el gráfico a generar por medio del asistente para gráficos de Maxim, este nos brinda las posibilidades de asignarle un nombre al gráfico y a cada uno de las series seleccionadas en una hoja electrónica, asignarle un rotulo al eje de las X o al eje de las Y, seleccionar un tipo de gráfico entre los que se tienen, de tipo barra, de tendencia o de pastel.

5.2.2.2.15. Formulario para el módulo Análisis Uni – Modelo

5.2.2.2.15.1. Formulario para modelos de seguros

The screenshot shows a software window titled "Modelos De Seguros" with two tabs: "Seguros Discretos" (selected) and "Seguros Continuos". The interface is divided into several sections:

- Seleccionar Tabla:** A table with columns "Nombre de tabla" and "Edad Ma". Below the table are buttons "***" and "Exportar Tabla".
- Características del modelo:** A form with the following fields:
 - Dirección de Base: [Empty text box]
 - Nombre de Tabla: [Empty text box]
 - Tipo de Indemnización: [Constante Discreta] (dropdown menu)
 - Periodo para cancelar: [Anual] (dropdown menu)
 - Momento de pago: [Al finalizar el periodo de t/q] (dropdown menu)
 - Edad: [Empty text box]
 - Vigencia: [Empty text box]
 - Diferimiento: [Empty text box]
 - Tasa Técnica: [Empty text box]Buttons: "Generar Modelo" and "Nuevo".
- Calcular:** Checkboxes for "Valor Actuarial" and "Riesgo", and a "Calcular" button.
- Modelos generados:** A table with columns: "Codigo", "Tipo de Seguro", "Edad", "Vigencia", "Diferido", "Tasa", and "Tipo de Indemnización".
- Bottom:** Buttons "Eliminar Modelo", "Eliminar Tabla", "Aceptar", and "Cancelar".

Figura 5.36. Formulario para modelos de seguros

Esta es la interfaz visual del modulo 1 de Maxim, el cual permite la generación de modelos individuales actuariales de seguros continuos o discretos de una manera flexible.

Antes de definir las características de un modelo debemos asignarle una ley de mortalidad si el modelo es continuo, o una tabla de mortalidad si el modelo es discreto, las tablas de mortalidad deben estar contenidas en el manejador de bases Microsoft Access 98 o 2000.

Una vez definidas las características de un modelo de seguro, se procede a generarlo, el usuario puede generar tantos modelos como permita la memoria de la computadora donde se encuentre instalado Maxim.

5.2.2.2.15.2. Formulario para modelos de rentas

The screenshot shows a software window titled "Modelos Discretos de Rentas". It contains several input fields and buttons for configuring a model. The "Características del modelo" section includes fields for "Dirección de Base", "Nombre de Tabla", "Tipo de Pago", "Periodo de Pago", "Pago De Cuota", "Edad", "Vigencia", "Diferimiento", and "Tasa Técnica". Below this is a "Calcular" section with checkboxes for "Valor Actuarial" and "Riesgo", and a "Calcular" button. At the bottom, there is a table titled "Modelos generados" with columns for "Codigo", "Edad", "Vigencia", "Diferido", "Tasa", "Tipo de Tabla", "Fracción", and "Periodo". The table is currently empty. Buttons for "Eliminar Modelo", "Eliminar Todos", "Aceptar", and "Cancelar" are located at the bottom of the window.

Figura 5.37. Formulario para modelos de rentas

Permite generar los modelos discretos de rentas asignados a una tabla de mortalidad, este formulario nos brinda la flexibilidad de asignarle las características que el usuario requiere a un modelo y después generarlo, una vez generado las características principales del modelo se detallan en “Modelos generados”.

5.2.2.2.15.3. Formulario para sensibilizar modelos de seguros y rentas

The image shows a software dialog box titled "Sensibilizar Modelos de seguros". It is divided into several sections:

- Modelos Generados:** An empty list box on the left.
- Modelos a sensibilizar:** An empty list box on the right.
- Navigation:** Two arrow buttons (right and left) positioned between the two list boxes.
- Carácterística a considerar:** A dropdown menu currently showing "Tasa Técnica" and a text input field containing "Tasa Técnica".
- Range:** Three input fields labeled "Desde", "Hasta", and "Incremento", all currently empty.
- Sensibilizar:** Two radio buttons, "Valor Actuarial" and "Riesgo", both of which are unselected.
- Buttons:** "Aceptar" and "Cancelar" buttons at the bottom right.
- Descripción del modelo:** A large, empty text area at the bottom of the dialog.

Figura 5.38. Formulario para sensibilizar los modelos

Una vez generado los modelos de seguros o rentas estos pueden ser sensibilizados variando algunas de sus características como la tasa de interés, la edad, la vigencia o el diferimiento, a su vez al seleccionar el código del modelo dentro de las listas de “modelos generados” sus características se presentan en “Descripción del modelo”.

5.2.2.2.16. Formulario para el módulo Análisis Multi – Modelo

5.2.2.2.16.1. Formulario para iniciar un plan

The image shows a software window titled "Form1" with a purple title bar. The window is divided into two main sections. The left section, titled "Seleccionar Tabla", contains a list box with the text "tblTablas" and a button labeled "Escoger Tabla". The right section, titled "Características del Plan", has two tabs: "Plan de Obligaciones" and "Plan de Pagos". Below the tabs are three input fields labeled "Dirección", "Tabla", and "Tasa". Underneath these are two more input fields labeled "Intervalo Inf" and "Intervalo Sup", and a list box labeled "LstIntervalosFP". At the bottom of the right section are two buttons labeled "Elim" and "Sig". At the very bottom of the window are two buttons labeled "Aceptar" and "Cancelar".

Figura 5.39. Formulario para inicio de un plan

Esta forma ofrece la posibilidad de inicializar los planes de obligaciones y pagos antes de generarlos, la forma de inicializar un plan consiste en asignarle una tabla de mortalidad, la tasa de interés y los límites superiores e inferiores para cada intervalo que contenga el plan.

5.2.2.16.2. Formulario para generar plan de obligaciones

The screenshot shows a software window titled "Plan de obligaciones" with a purple title bar. It has two tabs: "Seguro" and "Renta", with "Renta" selected. The window is divided into several sections. The "Intervalo" section contains two radio buttons: "Con modelo" (selected) and "Sin modelo". The "Rentas" section contains three rows of input fields: "Tipo de Pago" with a dropdown menu set to "Constante" and a "C" button; "Periodo de Pago" with a dropdown menu set to "Anual"; and "Pago De Cuota" with a dropdown menu set to "Vencido". A "Generar" button is located below these fields. The bottom section, "Modelos del Plan de Obligaciones", features a "SigMod" button and a table with four columns and one visible row. At the bottom of the window are buttons for "Eliminar", "Cancelar", and "Aceptar".

Figura 5.40. Formulario para generar plan de obligaciones

Al definir los límites de los intervalos en cada uno de estos se genera un modelo actuarial de seguro o renta como parte del plan de obligaciones este formulario nos brinda esta posibilidad como se observa en la Fig. 5.40. Las características de los modelos generados como parte de este plan se describe en “Modelos del Plan de Pagos”

5.2.2.2.16.3. Formulario para el plan de pagos

The screenshot shows a software window titled "Form1" with a purple title bar. It contains several sections:

- Intervalo:** A group box containing two radio buttons: "Con modelo" (selected) and "Sin modelo".
- Rentas:** A group box containing three dropdown menus: "Tipo de Pago" (set to "Constante"), "Periodo de Pago" (set to "Anual"), and "Pago De Cuota" (set to "Vencido"). A small button labeled "C" is to the right of the first dropdown. A "Generar" button is located below these fields.
- Modelos del Plan de Pagos:** A group box containing a "Sig" button and a table with four columns and one visible row. The table has a vertical scrollbar on the right and a horizontal scrollbar at the bottom.
- Buttons:** At the bottom of the form are three buttons: "Eliminar", "Aceptar Plan", and "Cancelar Plan".

Figura 5.41. Formulario para el plan de pagos

En esta forma se definen los modelos actuariales de renta para cada intervalo descrito en la forma “Iniciar plan” como parte del plan de pagos, las características de los modelos generados como parte de este plan se describe en “Modelos del Plan de Pagos”

5.2.2.2.16.4. Formulario para calcular primas y reservas

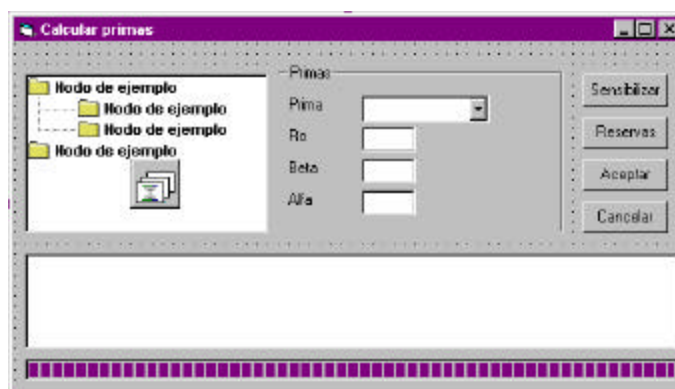


Figura 5.42. Formulario para calcular primas y reservas

Esta forma nos brinda la posibilidad de observar las características que fueron definidas por el usuario para el plan de obligaciones y pagos y a su vez calcular las primas netas, comerciales y de inventario así como las reservas matemáticas.

5.2.2.3. Otros módulos estándar y de clase

Maxim posee módulos estándar y de clase que no forman parte de los modelos de objetos ya definidos, pero que manipulan los objetos que estos generan para alcanzar un fin específico.

5.2.2.3.1. Módulo estándar para la manipulación de hojas

5.2.2.3.1.1. Procedimiento: InsertarNuevaHoja

Este procedimiento permite insertar una hoja nueva en la colección de hojas, la presenta visualmente en el sistema y de manera representativa en el administrador de hojas y gráficos.

5.2.2.3.1.2. Procedimiento: InsertarHojaAbierta

Este procedimiento permite insertar una hoja existente en la colección de hojas, la presenta visualmente en el sistema y de manera representativa en el administrador de hojas y gráficos.

5.2.2.3.1.3. Procedimiento: EliminarHoja

Este procedimiento permite eliminar la hoja seleccionada en el administrador de hojas y gráficos, removiéndola de la colección de hojas. Antes de eliminarla completamente se invoca al procedimiento **PreguntarPorGuardar**.

5.1.1.3.1.4 Procedimiento: PreguntarPorGuardar

Este procedimiento permite al usuario guardar una hoja que se va eliminar, asignándole un nombre a esta, o guardar los cambios de una hoja que ya fue antes abierta.

5.2.2.3.1.4. Procedimiento: InsertarNombreNuevoHoja

Este procedimiento permite asignar un nombre nuevo a la hoja seleccionada en el administrador de hojas y gráficos.

5.2.2.3.2. Módulo estándar para la manipulación de gráficos

5.2.2.3.2.1. Procedimiento: AñadirGráficoArbolCole

Este procedimiento asigna un objeto gráfico a la hoja activa y lo añade en la colección de gráficos y en el administrador de hojas y gráficos.

5.2.2.3.2.2. Procedimiento: BorrarGráfico

Este procedimiento elimina el gráfico seleccionado de su colección y del administrador de hojas y gráficos

5.1.1.1.1 Módulo de clase para primas y reservas

5.1.1.1.1 Propiedad: PlanDeObligacionesSeg

Esta propiedad contiene una colección de objetos con los modelos de seguros discretos generados para el plan obligaciones

5.1.1.1.2 Propiedad: PlanDeObligacionesRent

Esta propiedad contiene una colección de objetos con los modelos de rentas discretas generadas para el plan de obligaciones

5.1.1.1.3 Propiedad: PlanDePagos

Esta propiedad contiene una colección de objetos con los modelos de rentas discretas generadas para el plan de pagos.

5.1.1.1.4 Propiedad: TasaPlan

Esta propiedad contiene la tasa de interés asignada tanto para el plan de obligaciones como de pagos.

5.1.1.1.1.5 Propiedad: TablaPlan

Esta propiedad contiene la tabla de mortalidad asignada tanto para el plan de obligaciones como de pagos.

5.1.1.1.1.6 Método: CalcularPrimaNeta

Este método calcula la prima neta considerando los planes generados, y para la tasa de interés y tablas asignadas por el usuario.

5.1.1.1.1.7 Método: CalcularPrimaComercial

Este método calcula la prima comercial considerando los planes generados, para la tasa de interés y tablas asignadas por el usuario, tomando en cuenta los gastos internos y externos del plan.

5.1.1.1.1.8 Método: CalcularPrimaInventario

Este método calcula la prima de inventario considerando los planes generados, para la tasa de interés y tablas asignadas por el usuario, tomando en cuenta los gastos internos y externos del plan

CAPITULO VI

6. LA HOJA ELECTRÓNICA ACTUARIAL EN TIEMPO DE EJECUCIÓN

En este capítulo se detallará el último paso del ciclo de vida de un sistema, la implementación y pruebas de Maxim, mencionados en el capítulo 5, se mostrará los resultados obtenidos de la implementación, además de ciertas pantallas donde se generan los modelos actuariales y se consideran los requerimientos del usuario.

Maxim es una aplicación que fue desarrollada bajo Windows 98, pero puede ser implementada también en Windows 2000 o Windows

Milenium, los requisitos mínimos de hardware para un desempeño aceptable de Maxim son los siguientes: procesador de 500 MHZ, 32 MB de memoria, espacio mínimo en disco duro de 20 MB, monitor VGA.

6.1. Inicio de Maxim

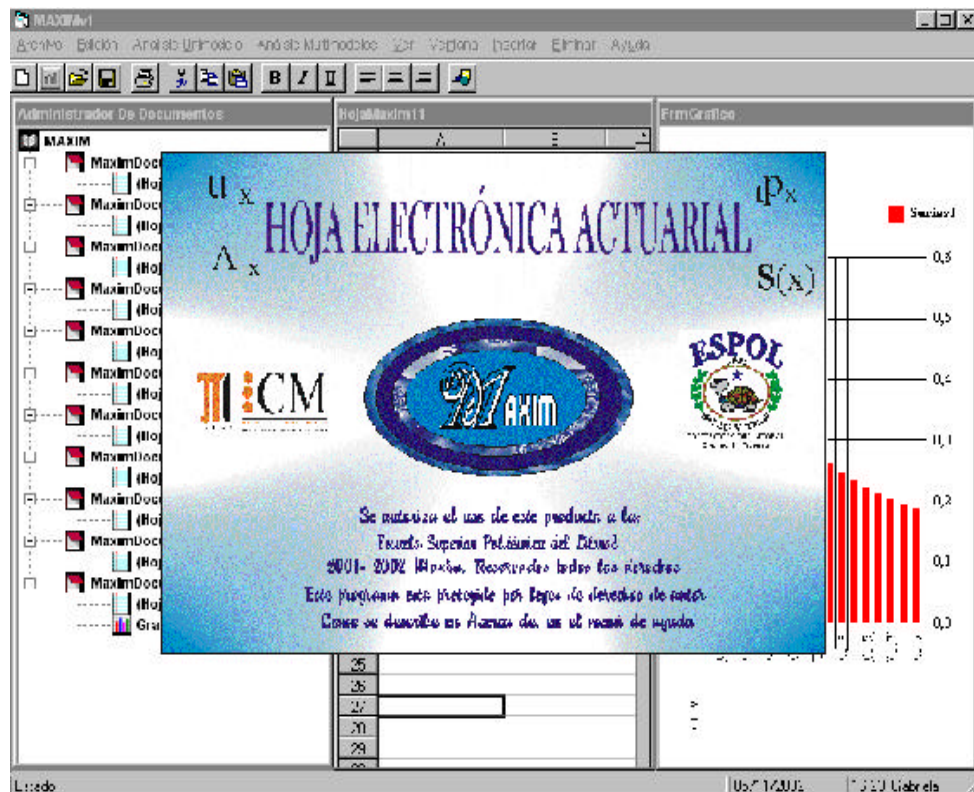


Figura 6.1 Pantalla inicial de Maxim

El instalador de Maxim crea un acceso directo dentro del escritorio, de modo que el usuario puede cargar la aplicación desde este sitio, los

componentes de código se registran de forma automática en la carpeta REGYSTER del sistema, así distintas aplicaciones diferentes a Maxim que admitan automatización Active X pueden hacer uso de Seguros.dll y Rentas.dll.

La figura 6.1. muestra la portada de Maxim al iniciar la aplicación, además de las características visuales más relevantes del sistema, como el objeto gráfico, el objeto hoja electrónica y el administrador de hojas y gráficos.

6.2. El asistente para gráficos

Como se mencionó en el capítulo 5 Maxim posee un asistente gráfico, el cual brindará al usuario una herramienta visual para el análisis de los resultados.

6.2.1. Escoger entre tres tipos de gráficos



Figura 6.2. Asistente Gráfico selección de gráficos

En la figura 6.2 se presenta la interfaz visual inicial del asistente gráfico de Maxim, aquí el usuario podrá seleccionar entre tres tipos de gráficos diferentes, el que más se ajuste a sus necesidades, como por ejemplo el gráfico de barras, el gráfico para series de tiempo, o el gráfico de pastel.

Para activar el asistente gráfico de Maxim el usuario tuvo que seleccionar por lo menos una serie con datos numéricos en la hoja de cálculo activa.

6.2.2. Series del gráfico

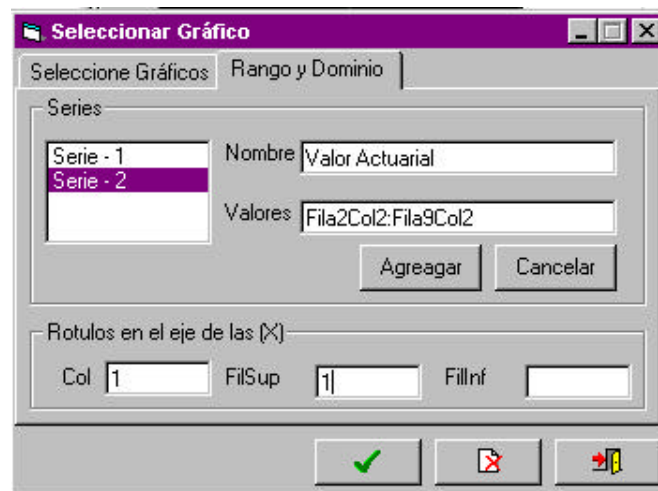


Figura 6.3. Asistente Gráfico rango y dominio

Una vez seleccionado el tipo de gráfico, el usuario tiene la posibilidad de asignarle un nombre a cada serie seleccionada en la hoja de cálculo, sólo se debe seleccionar la serie y agregarle un nombre, por omisión se mostrará en “Valores” el rango de la serie seleccionada, como se observa en la figura 6.3.

6.2.3. Rótulos en los ejes X e Y

Este mismo asistente brinda la oportunidad de especificar un rango de etiquetas para el eje de las X, así se observa en la parte baja de la figura 6.4 “Rótulos en el eje de las (X)”

6.3. El administrador de Hojas y Gráficos

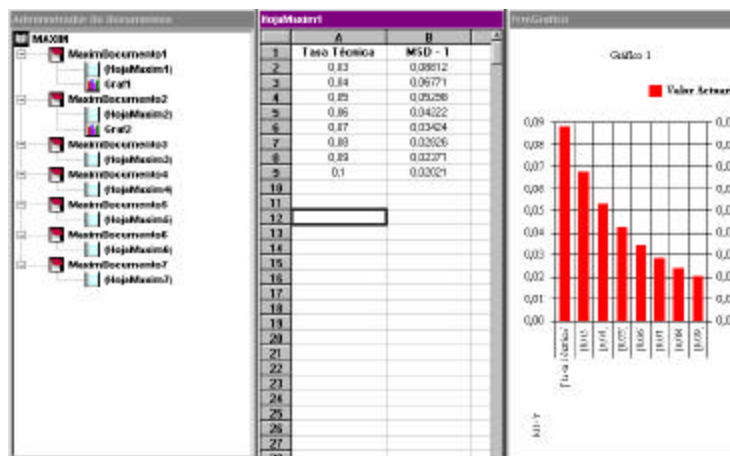


Figura 6.4. Administrador de hojas y gráficos

Maxim ofrece la oportunidad al usuario de trabajar con una gran cantidad de hojas electrónicas abiertas y gráficos pertenecientes a estas hojas y mantenerlos organizados dentro del Administrador de hojas y gráficos, como se observa en la parte derecha de la figura 6.4.

El administrador de gráficos y hojas de Maxim presenta el estado actual de hojas abiertas, además muestra cuantas de éstas hojas tienen anexo un gráfico, a parte de esto da la posibilidad de ir entre hojas y observar su contenido o eliminar una hoja que el usuario ya no requiera.

El administrador presenta 4 nodos, los cuales representan los elementos visuales que componen la hoja electrónica, el primer nodo denominado "MAXIM" representa la aplicación entera y es el nodo principal o padre, el segundo nodo es el nodo "MaximDocumento" el cual contiene una hoja electrónica y el gráfico perteneciente a la hoja (si ésta tuviera uno), el tercer nodo es el nodo "HojaMaxim" y representa a una hoja electrónica, el cuarto nodo es el nodo "GráficoMaxim" y representa un gráfico. Estos son los nombres que se asignan originalmente al crear uno de estos componentes, pero el usuario los puede cambiarlos si así lo desea con la excepción del nodo "MAXIM" y el nodo "MaximDocumento".

6.3.1. Insertar una hoja nueva

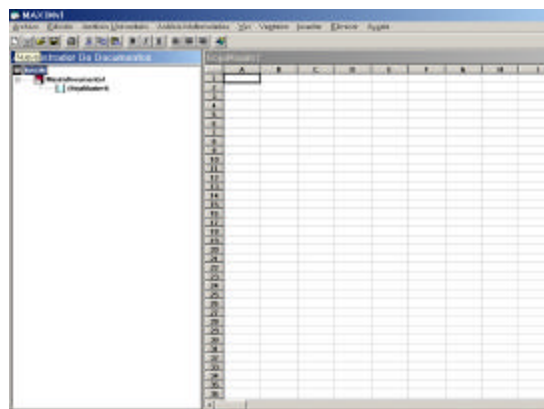


Figura 6.5 Hoja Insertada

El usuario podrá añadir hojas a Maxim por dos vías, la primera dirigiéndose al menú archivo y después seleccionar Nuevo, la segunda pulsando el botón nuevo de la barra estándar, ésta hoja se representará dentro del administrador de gráficos y hojas, como se observa en la figura 6.5.

6.3.2. Guardar una hoja en disco

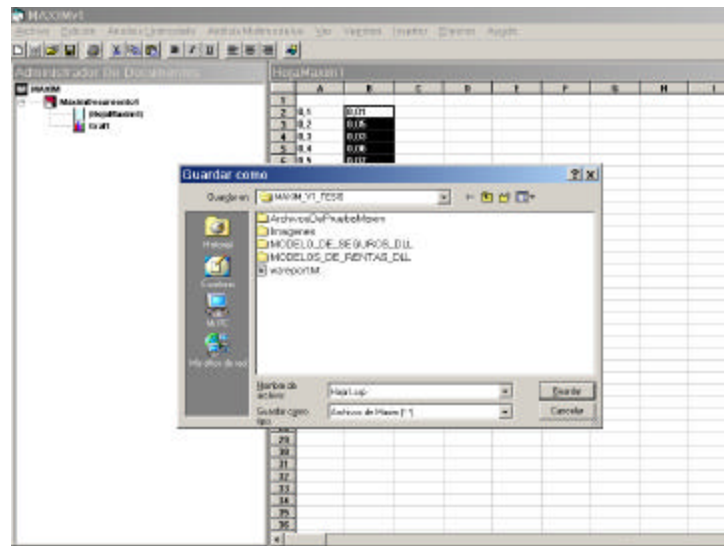


Figura 6.6. Guardar una hoja

Las hojas electrónicas se guardan en el disco de manera individual, esto permite añadir cualquier hoja anteriormente editada

al administrador de hojas y gráficos para así comenzar a trabajar con ésta, conjuntamente con las que se encuentran ya abiertas.

6.3.3. Insertar un gráfico

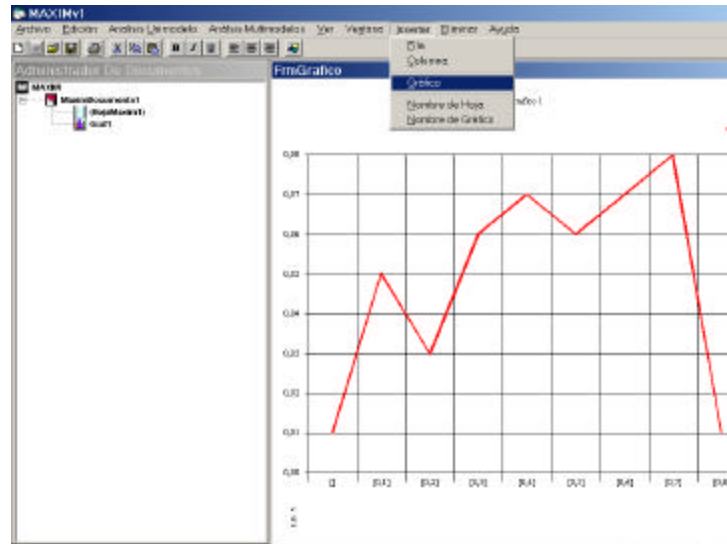


Figura 6.7. Insertar un Gráfico

Una hoja puede poseer un gráfico anexo, éste se lo puede insertar por dos vías, la primera por el menú Insertar, aquí seleccionamos gráfico, como se observa en la figura 6.7, la segunda activando el segundo botón, de derecha a izquierda de la barra estándar.

Para tener la posibilidad de obtener un gráfico debe de haber una hoja activa y dentro de ésta un rango numérico de celdas seleccionadas.

Al insertar un gráfico, éste se lo representa por un nodo dentro del administrador de hojas y gráficos, como se observa en la figura 6.8, de ésta manera activamos el gráfico requerido sólo pulsando el nodo gráfico que lo representa.

6.3.4. Eliminar un gráfico

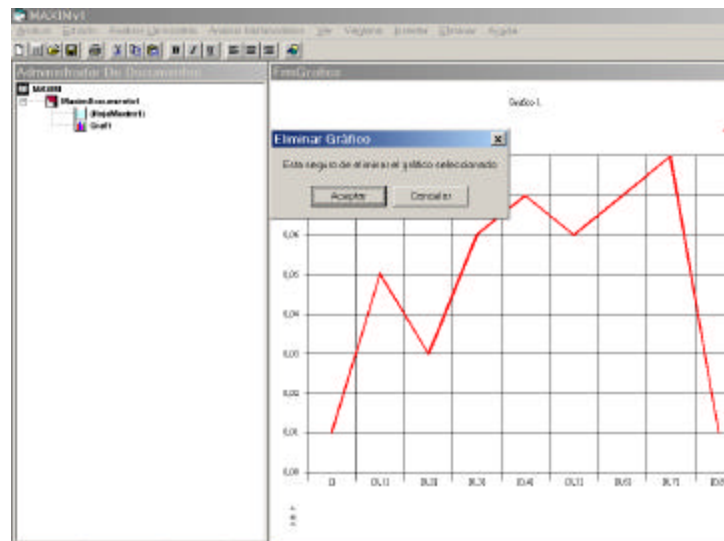


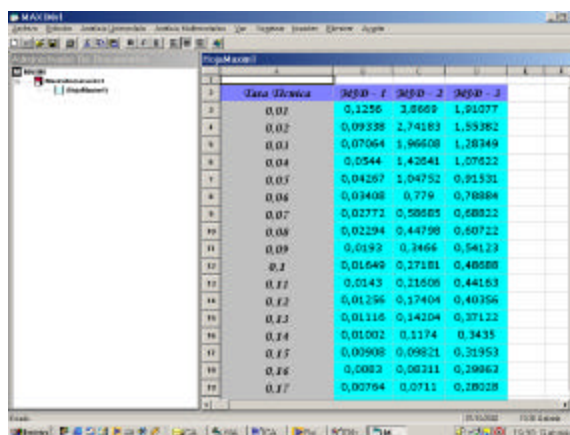
Figura 6.8. Eliminar un gráfico

Al pulsar dos veces un nodo gráfico en el administrador este se elimina de memoria y a su vez del administrador, pero antes de esto Maxim solicita una confirmación de la eliminación como se observa en al figura 6.8.

6.3.5. Moverse entre hojas y gráficos

El administrador de hojas y gráficos permite la flexibilidad necesaria para movernos entre las hojas abiertas y ver el contenido de éstas así como los gráficos que tienen anexados. De esta manera el usuario podrá organizar los resultados en distintas hojas dependiendo del análisis y compararlos moviéndose entre las hojas y gráficos que competan.

6.4. El menú de edición



	Tasa de crecimiento	2000 - f	2000 - 2	2000 - 3
2000				
2001	0.02	0.1256	3.8663	1.01077
2002	0.02	0.09338	2.74183	1.53382
2003	0.01	0.07064	1.96608	1.28349
2004	0.04	0.0544	1.42041	1.07622
2005	0.03	0.04267	1.04752	0.91331
2006	0.04	0.04408	0.779	0.76884
2007	0.07	0.02772	0.58885	0.68822
2008	0.08	0.02294	0.44798	0.60722
2009	0.09	0.0193	0.3466	0.54123
2010	0.1	0.01645	0.27181	0.48588
2011	0.11	0.0143	0.21606	0.44183
2012	0.12	0.01256	0.17404	0.40356
2013	0.13	0.01118	0.14204	0.37122
2014	0.14	0.01002	0.1174	0.3435
2015	0.15	0.00908	0.09821	0.31953
2016	0.16	0.0083	0.08211	0.29863
2017	0.17	0.00764	0.0711	0.28028

Figura 6.9. Edición para las hojas electrónicas

El usuario podrá personalizar la presentación de los resultados contenidos en cada una de las hojas como se observa en la figura 6.9, de ésta forma no se estará ligado a reportes fijos, con colores fijos o tipos de letras fijas, si no que este diseñará un reporte de acuerdo a su preferencia.

6.4.1. Dar formato al contenido de una hoja de cálculo

Una celda puede contener cualquier tipo de dato, la herramienta que ofrece Maxim para dar formato al contenido de las hojas son las estándar existentes en cualquier hoja de cálculo, así tenemos que el usuario puede asignar cualquier tipo de letra de las definidas en la PC, ajustar el tamaño de la fuente o darle un color específico.

6.4.2. Dar formato a las celdas de una hoja de calculo

El usuario puede personalizar y diferenciar cada área de datos en una hoja, asignándole un color como se observa en la figura 6.10.

6.5. Módulo para el análisis uni – modelo

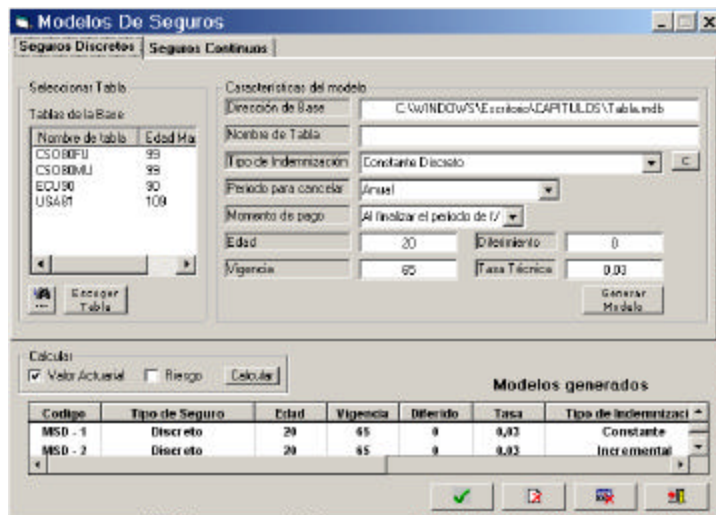


Figura 6.10. Interfaz visual para generar modelos

La figura 6.10 muestra la interfaz visual de uno de los componentes del módulo 1 de Maxim en tiempo de ejecución, se observa en la parte izquierda la sección que permite al usuario seleccionar una tabla, al presionar el botón “Escoger Tabla” se asigna una tabla de mortalidad a un modelo actuarial.

En “Características del modelo” se procede a personalizar el modelo que se quiere generar, al presionar el botón “Generar Modelo” se procede a generar el modelo con las cualidades definidas por el usuario.

En la parte baja de la figura 6.10 se observa todos los modelos generados por el usuario identificados por su código (MSD para modelos discretos, MSC para modelos continuos) , en éste nivel el usuario puede editar el número de modelos generados, eliminando un modelo específico o todos. A continuación presentaremos varios resultados brindados por el sistema para modelos continuos y discretos, pero antes presentamos la siguiente tabla.

Tabla IV

Las tablas de mortalidad asignadas a los modelos

Tabla	Descripción	Fuente
ECU90	Tabla de mortalidad de Ecuador para hombres y mujeres	INEC, Censo de 1990
CSO80 MU	Tabla de mortalidad masculina del año de 1980 para U.S.A	Seminario de Swiss Re Life & Healt dictado en Colombia
CSO80FU	Tabla de mortalidad femenina del año 1980 para U.S.A	Semiario de Swiss Re Life & Healt dictado en Colombia

La tabla IV muestra los nombres de las tablas de mortalidad consideradas para la generación de los modelos actuariales.

Se va a considerar para la ejecución del módulo análisis Unimodelos los siguientes modelos de seguros discretos y continuos

Tabla V
Modelos discretos considerados

Código del modelo	Tabla	Descripción
MSD – 1	CSO80 FU	Modelo con indemnización constante para una persona de edad 20 años temporal hasta 40 años más.
MSD – 2	CSO80 MU	Modelo con indemnización constante para una persona de edad 20 años temporal hasta 40 años más
MSD – 3	CSO80 FU	Modelo con idemnización Incremental, para una persona de 20 años de edad temporal hasta 40 años más
MSD – 4	CSO80 MU	Modelo con idemnización Incremental, para una persona de 20 años de edad temporal hasta 40 años más
MSD – 5	ECU90 MU	Modelo con indemnización constante para una persona de edad 20 años temporal hasta 40 años más.

La tabla V define los códigos de los modelos, así como las tablas de mortalidad asignada a cada uno, por último tenemos una breve descripción de las características de los modelos generados.

Tabla VI

Modelos continuos considerados

Código del modelo	Ley	Descripción
MSC – 1	Ley Constante	Modelo continuo para una persona de 20 años temporal 40 años más con indemnización constante
MSC – 2	Ley de Gompertz	Modelo continuo para una persona de 20 años temporal 40 años más, con indemnización logarítmica
MSC – 3	Ley de Moivre	Modelo continuo para una persona de 20 años temporal 40 años más, con indemnización cuadrática.

Quando se haga referencia a un modelo actuarial se lo realizará en base a su código, así mismo al referirse a las tablas actuariales se considerarán los nombre de las tablas definido en la tabla I.

6.5.1. Generación de Seguros Discretos

Se generaron los modelos descritos en la Tabla II, por medio de la interfaz visual de la figura 6.11, para así obtener los valores actuariales de estos, se han considerado tres modelos con indemnización constante (MSD-1,MSD-2,MSD-5) pero asignados

a diferentes tablas de mortalidad, lo mismo ocurre con los modelos cuya indemnización es incremental.

Todos los resultados presentados a continuación fueron calculados con una indemnización unitaria y a una tasa del 3%.

Tabla VII

Valores actuariales para los modelos de seguros discretos

Código del modelo	Valor Actuarial
MSD – 1	0,05727
MSD – 2	0,07917
MSD – 3	1,38986
MSD – 4	1,91162

Los modelos asignados a la tabla CSO80 FU son los que poseen un valor actuarial menor ($MSD - 1 < MSD - 2$ y $MSD-3 < MSD-4$) como se observa en la tabla VII, recordemos que CSO80 FU es la tabla que representa a las mujeres de U.S.A en 1980, mientras que CSO80 MU representa a los varones de U.S.A en el mismo año, como el componente financiero es el mismo para todos los modelos, esto es una tasa técnica del 3 % entonces la razón de este desnivel se debe principalmente a la estructuras de las tablas

de mortalidad, así pues se puede concluir que las mujeres en 1980 poseían un índice de mortalidad menor al de los hombres, reflejándose ésta característica en el valor actuarial de los modelos.

6.5.2. Generación de Seguros Continuos

La generación de modelos para seguros continuos al igual que los discretos se los realiza mediante la interfaz presentada en la figura 6.11, pero a diferencia de los discretos donde se puede tener una gran cantidad de tablas de mortalidad, los continuos sólo se les pueden asignar las leyes teóricas de mortalidad más conocidas, los modelos continuos que se generaron tienen asignados una de éstas leyes como se observa en la tabla III, además de indemnizaciones funcionales como las logarítmicas o las cuadráticas.

El objetivo principal de ésta sección es observar como varía el valor actuarial frente a cambios en los parámetros de las leyes de mortalidad, la tasa utilizada es del 3 %, con una indemnización unitaria.

Tabla VIII

Tabla de valores actuariales para el modelo MSC –1

Constante	Valor Actuarial
0,04	0,53946
0,03	0,4572
0,02	0,34797
0,01	0,20084

Para observar como varía el valor actuarial a cambios en los parámetros de las leyes de mortalidad primero, se escogió el modelo MSC –1 descrito en la tabla III, este es un modelo que tiene asignado una ley analítica de mortalidad constante, en la tabla VIII se observa el valor actuarial de este modelo para varios niveles en la constante de la ley analítica, algo que se aprecia a simple vista es como el valor actuarial del modelo decrece, conforme decrece la constante de la ley, esto es algo lógico ya que al disminuir ésta constante la propensión a la muerte del conglomerado es menor, de ahí que el modelo capta ésta información y el valor actuarial de este también disminuye.

Aunque las leyes de mortalidad en todo el rango de vida de un conglomerado no se presentan de una forma constante (si no más bien en forma de una tina de baño, con pequeñas fluctuaciones), si suele presentarse constante por tramos, de manera más frecuente en las edades medias, entre los 30 – 60 años, de ahí la importancia de ésta ley.

Tabla IX

Valores Actuarial para el modelo MSC –2

B / C	1,01	1,02	1,03
0,01	0,2694	0,35258	0,44474
0,02	0,44044	0,53489	0,61861
0,03	0,55277	0,6378	0,7036

Para generar la tabla IX se utilizó el modelo MSC –2, el cual tiene asignado la ley de mortalidad de Gompertz, lo que se observa es que las variaciones en los parámetros en la ley de Gompertz hacen muy sensible el valor actuarial del modelo analizado, para cada parámetro se tomo un incremento de 0,01 tomando como base inicial $B = 0,01$ y $C = 1,01$, el parámetro de la ley de Gompertz que más sensibiliza el valor actuarial es B como se observa en la tabla IX.

Tabla X

**Valor Actuarial para un modelo asignado a una ley de Moivre,
para varias indemnizaciones funcionales**

Indemnización Funcional	Valor Actuarial
Constante	0,29325
Lineal creciente	4,73527
Cuadrática	112,9748
Raíz	1,08937
Logarítmica	0,74604

La tabla X nos presenta el valor actuarial para el modelo continuo MSC-3, éste tiene asignado la ley teórica de Moivre, para el calculo de ésta tabla no se consideró la variación de la edad máxima del conglomerado, el cual es el único parámetro de la ley analítica de Moivre, más bien se fijo este parámetro en 100 años (es decir $W = 100$) para así obtener el valor actuarial para los distintos tipos de indemnizaciones funcionales que Maxim ofrece para los modelos continuos.

Sin considerar el componente de la influencia de la ley analítica en los valores actuariales de los modelos, la tabla X deja ver dos detalles importantes que nos pueden llevar a conclusiones relevantes acerca de las indemnizaciones funcionales, el primer detalle se aprecia en el valor actuarial para la indemnización cuadrática, es bastante más grande comparada con las demás, esto es debido a los grandes monto que indemniza en el caso de f/q , como se observa en la siguiente tabla.

Tabla XI

Indemnización del modelo continuo asignado a una indemnización funcional cuadrática.

Intervalo de fallecimiento	Valor a indemnizar
[20-21)	1 u.m
[21-22)	4 u.m
[22-23) y Así hasta	9 u.m
[59-60)	1600 u.m

Los valores a indemnizar presentados en la tabla XI son aproximados, ya que si el ente f/q no en el extremo del intervalo si no en una fracción de este, la idemnización también se fracciona, ya que estos modelos están diseñados para que la indemnización

se cancele en el momento de f/q , por ejemplo sí se supone que el individuo fallece a los 20,5 años entonces este recibiría sólo $0,5^2 = 0,25$ u.m.

Al ver los montos a indemnizar en la tabla XI, se justifica el valor actuarial tan alto a cancelar para este caso, puesto que la función cuadrática crece mucho más rápido que las demás indemnizaciones funcionales.

Otro detalle importante de anotar ocurre con las indemnizaciones logarítmica y raíz, ambos son los valores actuariales más cercanos, pero la logarítmica es la menor de las dos, esto se debe principalmente a las características que estas presentan, ambas son crecientes y cóncavas hacia abajo, pero la indemnización logarítmica crece menos deprisa que la indemnización raíz, se puede decir entonces que las características de las indemnizaciones funcionales influyen decisivamente en el valor actuarial de los modelos continuos.

6.5.3. Análisis de sensibilidad de modelos de seguros

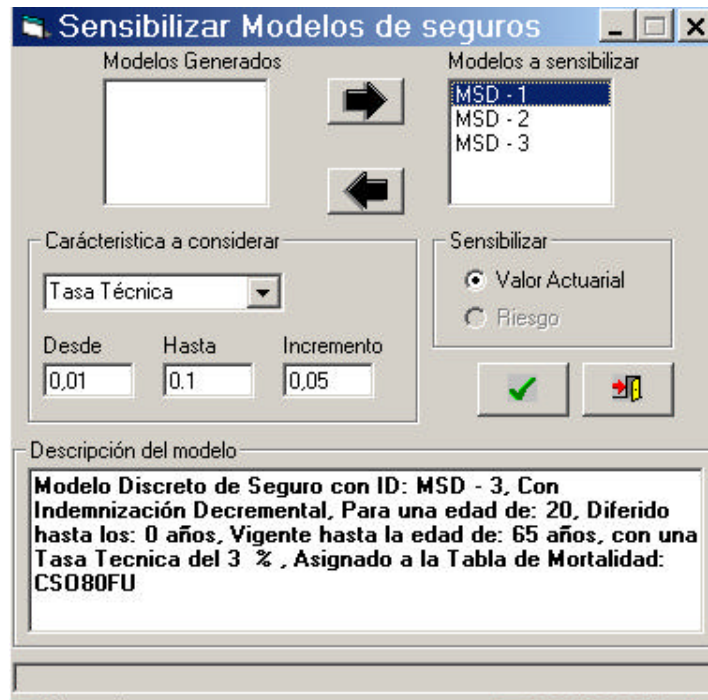


Figura 6.11. Interfaz visual para sensibilizar modelos de seguros

La figura 6.11 muestra el formulario para sensibilizar los modelos de seguros discretos y continuos, para llegar a este nivel el usuario antes tuvo que haber definido los modelos con los que quiere trabajar, estos modelos que se generaron son los que aparecen al iniciar el formulario en la sección “Modelos generados”, el usuario escogerá entre estos cuales desea sensibilizar y los colocará en la

sección “Modelos a sensibilizar” como se observa en la parte superior figura 6.12.

Si el usuario desea recordar que características tiene un modelo, éste debe de presionar sobre el código del modelo y de manera automática aparece la descripción en la sección “Descripción del Modelo”, como se aprecia en la parte baja de la figura 6.12, aquí se detallan todas las características del modelo, tales como la edad, la vigencia, la tabla o ley de mortalidad asignada etc. Esto sirve de ayuda al usuario al momento de decidir cual o cuales modelos desea sensibilizar.

El usuario tiene la opción de sensibilizar cuatro características comunes entre los modelos, tales como la edad del asegurado, la vigencia del modelo, el diferimiento del modelo y la tasa de interés.

6.5.3.1. Sensibilizar modelos de seguros a través de la tasa de interés

Tabla XII

Valores Actuariales de los modelos MSD-1, MSD-2, MSD-5 considerando distintas tasas de interés

Tasa Técnica	MSD – 1	MSD – 2	MSD – 5
0,03	0,05727	0,07917	0,07054
0,04	0,04559	0,06313	0,05612
0,05	0,03683	0,05115	0,04531
0,06	0,0302	0,04212	0,03712
0,07	0,02513	0,03523	0,03084
0,08	0,0212	0,02991	0,02598
0,09	0,01813	0,02576	0,02218
0,1	0,0157	0,02249	0,01917
0,11	0,01376	0,01987	0,01676
0,12	0,01219	0,01775	0,01482
0,13	0,01091	0,01602	0,01323
0,14	0,00985	0,01458	0,01191
0,15	0,00896	0,01338	0,01082
0,16	0,00822	0,01237	0,00989
0,17	0,00758	0,0115	0,00911
0,18	0,00704	0,01075	0,00843
0,19	0,00656	0,0101	0,00785
0,2	0,00615	0,00953	0,00734

Los valores actuariales que se muestran en la tabla XII fueron generados con los modelos discretos MSD – 1, MSD-2, MSD – 5 cada uno asignado a las tablas de mortalidad CSO80MU, CSO80FU, ECU90 respectivamente, la característica que se

consideró para sensibilizar los modelos fue la tasa de interés, del 3% hasta el 20% con un incremento de 1%.

Estos resultados nos muestra que aparte de sensibilizar los modelos Maxim nos da la oportunidad de comparar modelos asignados a distintas tablas de mortalidad.

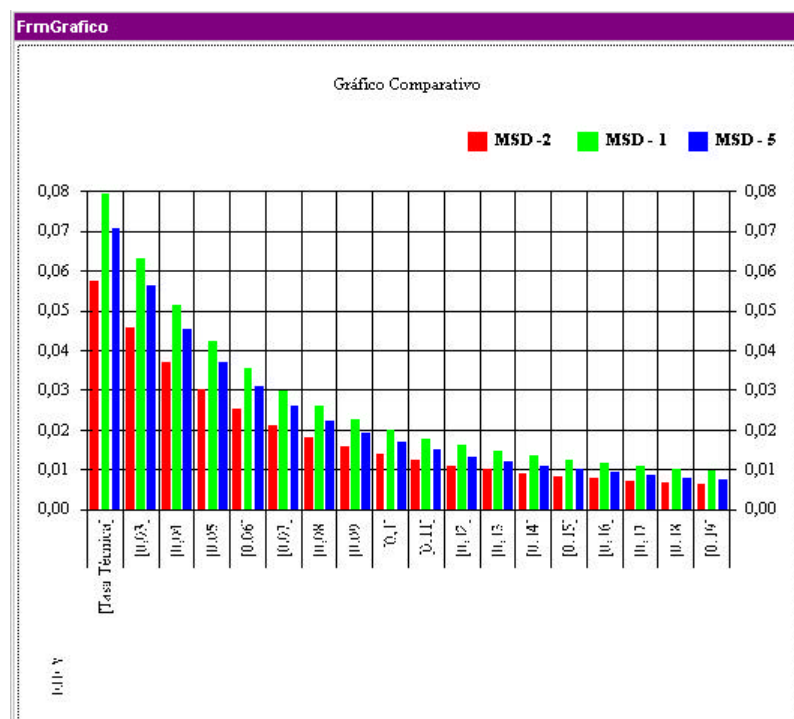


Figura 6.12. Gráfico de los valores actuariales para los modelos MSD-1, MSD-2, MSD-5 para varias tasas de interés

La figura 6.12 es un gráfico generado por Maxim basándose en los resultados de la Tabla XII se observa que mientras la tasa de interés crece el valor actuarial realiza el proceso inverso, además se puede ver lo sensible que son los modelos al variar ésta característica.

Los valores actuariales de los modelos decrecen de forma pronunciada y cóncava hacia abajo, sobre todas las tasas de interés se observa el mismo patrón de decrecimiento, el que presenta mayor valor actuarial es el modelo MRD-2 asignado a la tabla CSO80MU, seguido de MRD-5 asignado a la tabla ECU - 90 y por último MRD-1 asignado a la tabla CSO80FU, las mujeres siguen teniendo el menor valor actuarial.

Este análisis además nos sugiere que una empresa aseguradora si consigue en el mercado tasas más altas de interés que el de sus competidores estará en la capacidad de ofrecer productos más cómodos y competitivos económicamente para sus clientes.

6.5.3.2. Sensibilizar modelos de seguros a través de la edad del asegurado

Tabla XIII

Valores Actuariales para los modelos de seguros, variando la edad del asegurado

Edad	MSD - 2	MSD - 1	MSD -5
20	0,07054	0,05727	0,07917
25	0,07545	0,06095	0,08262
30	0,08069	0,06456	0,0874
35	0,08407	0,06765	0,09241
40	0,08487	0,06901	0,09552
45	0,08001	0,06587	0,09359
50	0,06902	0,05612	0,08271
55	0,04514	0,03604	0,05584
60	0	0	0

La tabla XIII fue generada considerando los mismos modelos que sirvieron para generar la tabla XII, se procedió a sensibilizar la edad del asegurado, que para los tres casos es de 20 años de edad, se sensibilizo hasta los 60 años puesto que los tres modelos son temporales hasta esta edad.

La tabla XIII nos muestra que el valor actuarial aumenta conforme la edad se acerca a la mitad del intervalo de 20 –60, después de este punto, mientras más se acerca a la edad limite de 60 años el valor actuarial disminuye, esto nos permite comparar y ver que

existen diferencias notable con los valores de la tabla XII, especialmente en el patrón de decrecimiento.

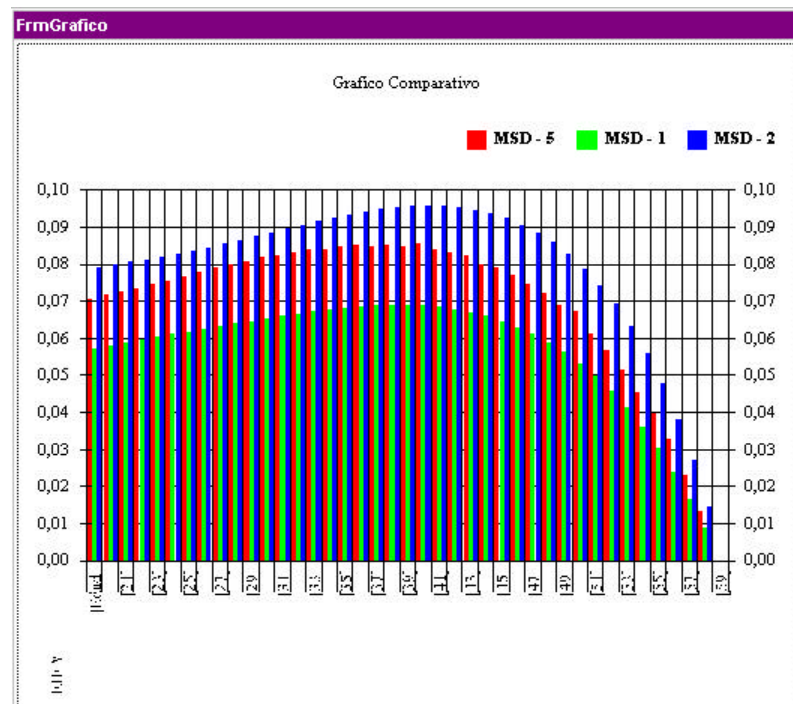


Figura 6.13 Gráfico de los valores actuariales para los modelos MSD-1, MSD-2,MSD-5 para distintas edades

La figura 6.13 muestra el patrón de ola que tiene el valor actuarial de los modelos al sensibilizar la edad, al llegar a la mitad del intervalo, el valor actuarial crece, para después caer de manera considerable hasta llegar a cero.

Como se observa en la figura 6.13, también predomina el mismo orden que en el análisis anterior, esto es MSC -2 es el mayor de todos, seguido de MSD-5 y por último MSD - 1, este particular que se presenta en el orden de los modelos se debe principalmente a las características diferentes de las tablas de mortalidad.

El resultado obtenido en la tabla XIII y la figura 6.14 trae una incógnita, esta es si ¿para todo modelo discreto al sensibilizar la edad presenta el mismo comportamiento descendente en su valor actuarial?. Bueno la respuesta a esto la obtendremos sensibilizando dos modelos adicionales a los presentados en las tablas II y III.

Tabla XVI

Modelos actuariales adicionales de dote puro y con indemnización incremental

Descripción del modelo	Tabla Asignada
Modelo incremental para un ente de 20 años de edad, temporal hasta los 60 años de edad	ECU 90
Modelo de dote puro para un ente de 20 años de edad temporal hasta los 60 años	ECU90

Se procede a sensibilizar estos dos modelos a través de la característica edad del asegurado, la tabla XV nos muestra los resultados de este proceso.

Tabla XV

**Valor Actuarial para los modelos de dote puro y con
Indemnización incremental.**

Edad	Incremental	Dote Puro
20	1,71422	0,25975
25	1,60266	0,30307
30	1,44447	0,3538
35	1,23801	0,41419
40	0,98706	0,48645
45	0,70159	0,57477
50	0,40717	0,68273
55	0,14542	0,81974
60	0	1

Como se ve el valor actuarial del modelo incremental decrece mientras que el modelo de dote puro aumenta conforme la edad se acerca más a la edad límite.

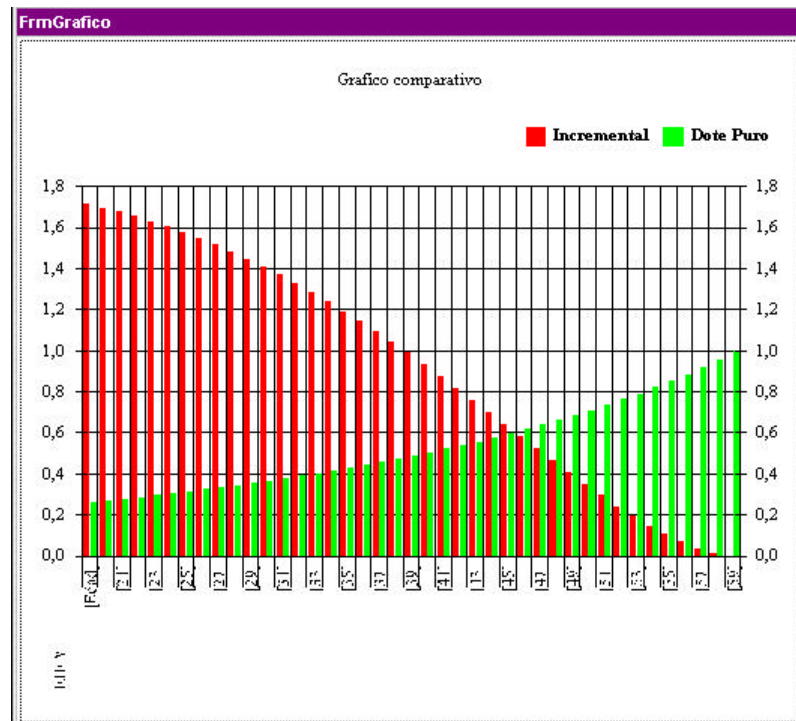


Figura 6.14 Gráfico de los valores actuariales para un modelo incremental y de dote puro, variando la edad

La figura 6.14 responde decisivamente a la incógnita planteada con anterioridad, el valor actuarial del modelo dote puro crece mientras la edad se acerca a los 60 años, además el valor final que toma es la unidad, esto es el valor de la indemnización, esto es algo normal si se considera que este modelo postula que se cancela la indemnización al asegurado siempre que el ente

sobreviva a un intervalo de tiempo, en este caso particular considerado que sobreviva a los 60 años de edad.

El modelo donde la indemnización en incremental decrece también, pero de una manera más decidida que los modelos MSD-1, MSD-2, MSD-5, además éste no presenta una forma de ola como los modelos presentados en la fig 6.13.

6.5.3.3. Sensibilizar modelos a través del diferimiento

Tabla XVI

**Valores actuariales y riesgos para los modelos se seguros,
variando los años de diferimiento**

Diferido	MSD – 1		MSD – 2		MSD – 3	
	VA	Riesgo	VA	Riesgo	VA	Riesgo
0	0,05727	0,02969	0,07917	0,04142	0,07054	0,03648
5	0,05229	0,02512	0,0706	0,03356	0,06467	0,03109
10	0,04749	0,02133	0,06387	0,02823	0,05924	0,02682
15	0,04261	0,018	0,05772	0,02403	0,05272	0,0224
20	0,03714	0,01479	0,05084	0,02	0,04532	0,01805
25	0,03014	0,01126	0,0422	0,01563	0,03616	0,01344
30	0,0217	0,00757	0,03132	0,01088	0,02626	0,00912
35	0,01168	0,0038	0,01751	0,00569	0,0143	0,00463

La tercera característica a sensibilizar es el diferimiento, para esto se selecciona nuevamente los modelos MSC – 1, MSC – 2, MSC –

5, ya que estos al tener idénticas características pero tablas de mortalidad diferentes, cumplen así los requisitos adecuados para realizar un análisis comparativo.

La tabla XVI fue generada variando a cada uno de los modelos los años de diferimiento desde el diferimiento inicial que es cero años, hasta los 35 años de diferimiento.

Se puede observar en la tabla XIII que los valores actuariales de los modelos disminuyen mientras los años de diferimiento aumentan.

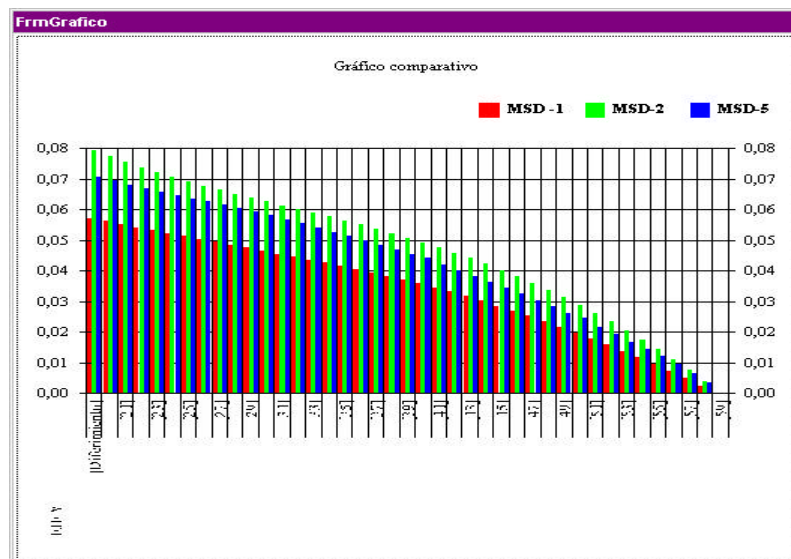


Figura 6.15 Gráfico de los valores actuariales para los modelos MSD-1, MSD-2, MSD-5 variando los años de diferimiento

Si comparamos las figuras 6.13, 6.14 y 6.15 nos muestra que no todas las características que han sido consideradas, sensibilizan de la misma manera a los modelos actuariales, en ésta última figura se observa que existe una tendencia decreciente, pero de una manera cóncava hacia arriba.

Sigue existiendo el dominio del modelo MSD -2 , puesto que este presenta el mayor valor actuarial en todos los años de diferimiento, el último lugar lo sigue ostentando el modelo MSD -1 , mientras que MSD-3 sigue manteniéndose en la mitad, ésta característica se ha mantenido en los tres tipos de análisis que se han realizado. Éste patrón se lo atribuyo especialmente a las características de las tablas de mortalidad, asignadas a cada modelo, unas tienen un índice de mortalidad más alto que otras, para confirmar esto, en la tabla XIII se tabuló el riesgo para cada modelo, para cada uno de los años de diferimiento.

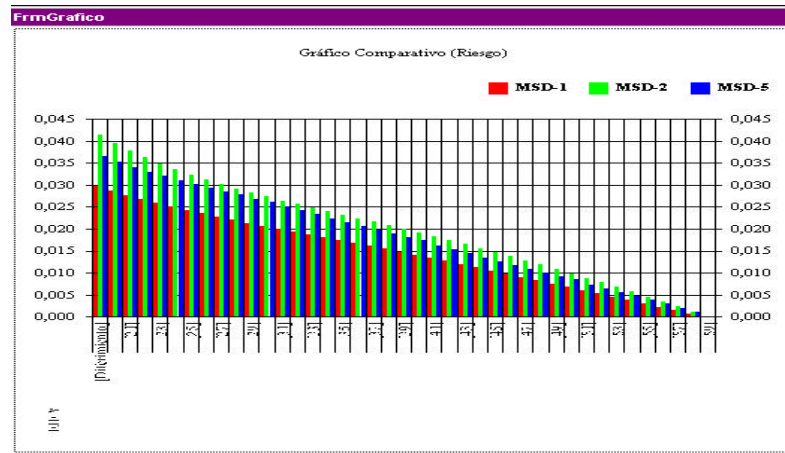


Figura 6.16 Gráfico del Riesgo para los modelos MSD-1, MSD-2, MSD-5

En la figura 6.16 se observa en forma clara el mayor riesgo que tiene MSD-2 de ahí que este presente el mayor valor actuarial de los tres modelos, seguido de MSD-5 y por último MSD-1, entonces el patrón jerárquico también se mantiene en el riesgo de los modelos.

6.5.3.4 Sensibilizar modelos a través de la vigencia

Tabla XVII

Valores Actuariales para modelos de seguros variando la edad de vigencia.

Vigencia	MSD - 1	MSD - 2	MSD - 3
25	0,00498	0,00857	0,00587
30	0,00978	0,01529	0,0113
35	0,01466	0,02145	0,01782
40	0,02013	0,02832	0,02522
45	0,02713	0,03696	0,03438
50	0,03557	0,04784	0,04428
55	0,04559	0,06165	0,05624
60	0,05727	0,07917	0,07054

La tabla XVII fue generada utilizando los modelos MSD -1, MSD-2, MSD-5 a cada uno de ellos se procedió a variar la edad de vigencia, fijada inicialmente en 60 años, iniciando con una vigencia de 25 años y culmina en los 60 años.

Los datos tabulados tienen una tendencia creciente en el valor actuarial, esto es mientras aumenta la edad de vigencia, aumenta el valor actuarial, este particular es contrario a los casos anteriores vistos.

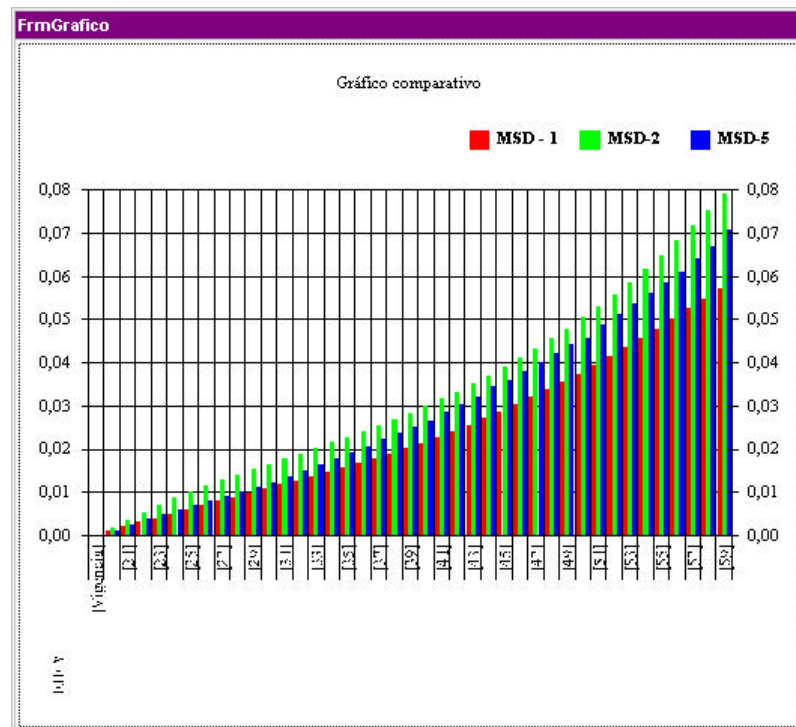


Figura 6.17 Gráfico para los valores actuariales de los modelos MSD-1, MSD-2, MSD-5, variando la vigencia del asegurado

La figura 6.17 muestra gráficamente el comportamiento de los modelos al sensibilizar la vigencia de estos, notemos que cuando la vigencia está muy cercana a la edad del asegurado el valor actuarial es casi nulo. Mientras la vigencia se aleja más de la edad del asegurado el valor actuarial crece de manera exponencial, esto es razonable puesto que mientras más grande es el intervalo de cobertura el riesgo de que el ente f/q dentro del intervalo es mayor.

6.5.4. Generación de modelos de rentas discretos.

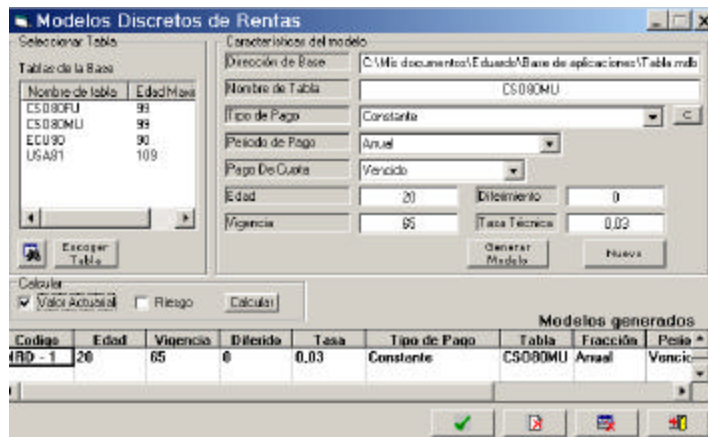


Figura 6.18. Interfaz visual para generar modelos de rentas discretos

La figura 6.18 muestra la interfaz visual para generar los modelos de rentas, al igual que en los modelos de seguros posee una sección denominada “Selección de tablas” donde se decide cual tabla de mortalidad se asignara al modelo de renta a generar, más a la derecha esta la sección de “Características del modelo”, donde el usuario personaliza el modelo de renta, al generar el modelo éste se lo representa en la sección “Modelos Generados”, en esta parte el usuario puede ver todas los atributos que este le asigno al modelo.

El usuario puede modificar el número de modelos generados, eliminando uno de éstos o todos según sean sus necesidades.

Los siguientes modelos de rentas son los que serán utilizados en el transcurso de la sección, además las tablas de mortalidad asignadas a los modelos son aquellas definidos en la tabla I.

Tabla XVIII

Modelos de renta discretos considerados

Código del modelo	Tabla	Descripción
MRD-1	CSO80FU	Modelo de renta con pagos anticipados constante para una persona de 20 años vigente hasta los 70
MRD-2	CSO80MU	Modelo de renta con pagos anticipados constante para una persona de 20 años diferido hasta los 70
MRD-3	CSO80FU	Modelo de renta con pagos anticipados cuya indemnización es incremental para una persona de 20 años diferido hasta los 70
MRD-4	CSO80MU	Modelo de renta con pagos anticipados cuya indemnización es incremental para una persona de 20 años diferido hasta los 70
MRD-5	CSO80MU	Modelo de renta con pagos anticipados fraccionarios para una persona de 20 años vigente hasta los 70 años

6.5.5. Análisis de sensibilidad de los modelos de rentas

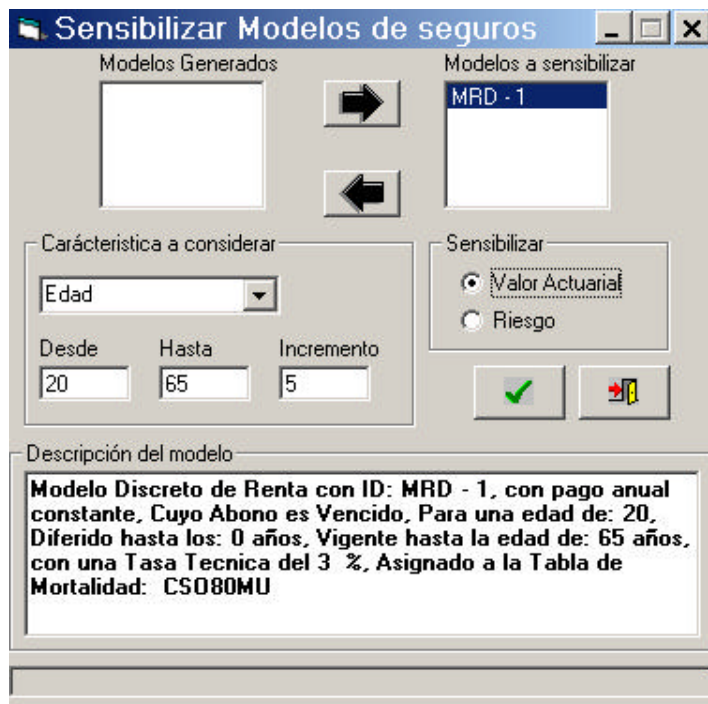


Figura 6.19 Interfaz de visual para sensibilizar modelos de rentas

Maxim da la oportunidad al usuario de sensibilizar los modelos de rentas generados, la figura 6.19 presenta la interfaz visual que nos permite lograr este fin, como se observa existen dos cuadros de listas el cuadro de lista de la izquierda presentan todos los modelos generados por el usuario, al pulsar el código de estos modelos en la parte baja del formulario, en “Descripción del

modelo” se puede apreciar las características con las que fue creado.

De los modelos generados, el usuario deberá seleccionar cuales de ellos desea sensibilizar, para que esto ocurra se deberá seleccionar el modelo requerido y pulsar el botón flecha que apunta hacia la derecha, automáticamente el código del modelo se ubicará en el cuadro de lista “Modelos seleccionados” como se aprecia en la figura 6.19.

Los seleccionados pueden ser sensibilizados por las mismas características que los modelos de seguros, estas son la tasa de interés, la edad del asegurado, los años de diferimiento y la vigencia del modelo.

A continuación presentamos los resultados brindados por Maxim para los modelos definidos en la tabla XVIII, todos los resultados fueron obtenidos con indemnización unitaria para modelos constantes y con base unitaria para modelos incrementales.

6.5.5.1. Modelos de renta sensibilizados a través de la tasa de interés

Tabla XIX

Valores actuariales para modelos de rentas discretos, variando la tasa de interés

Tasa Técnica	MRD - 1	MRD - 2
0,03	25,38987	24,94094
0,04	21,53549	21,20887
0,05	18,57465	18,33231
0,06	16,26221	16,07875
0,07	14,42716	14,28541
0,08	12,94868	12,83696
0,09	11,74037	11,65059
0,1	10,73962	10,66612
0,11	9,90048	9,83927
0,12	9,18883	9,13703
0,13	8,57898	8,53452
0,14	8,0514	8,01275
0,15	7,59102	7,55704
0,16	7,18613	7,15595
0,17	6,82749	6,80045
0,18	6,50774	6,48332
0,19	6,22097	6,19877
0,2	5,96241	5,94209
0,19	6,22097	6,19877

La tabla XIX fue generada haciendo uso de los modelos MRD-1, MRD-2, para esto se procedió a variar la tasa de interés desde el 3% hasta el 20% en incremento de un punto porcentual.

Se aprecia que mientras la tasa de interés aumenta, el valor actuarial de la renta disminuye, además se observa que el orden jerárquico que se mantenía en los modelos de seguros ya no se mantiene para los modelos de rentas, el modelo de renta asignado a la tabla CSO80FU es ligeramente mayor a los valores actuariales del modelo asignado a la tabla CSO80MU, para cada tasa de interés este resultado se aprecia mejor en la siguiente figura.

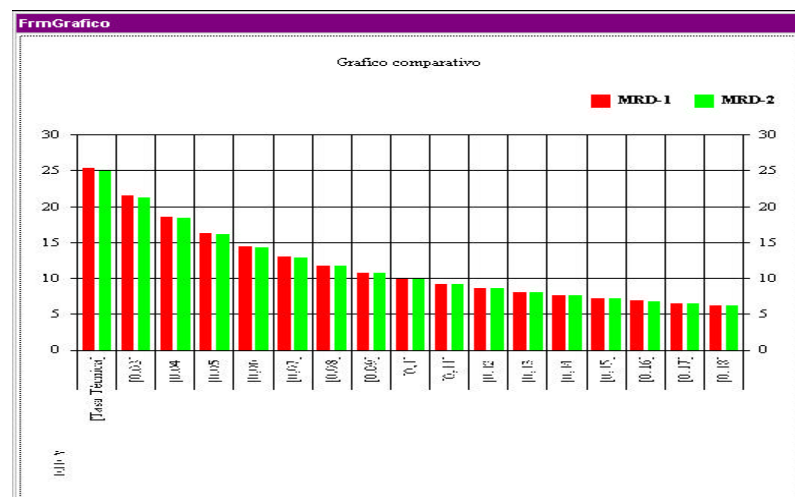


Figura 6.20 Gráfico de los valores actuariales para los modelos MRD-1, MRD-2, variando la tasa de interés.

La figura 6.21 muestra el decrecimiento cóncavo hacia abajo del valor actuarial al exponer los modelos a variaciones en la tasa de interés, este patrón es similar a los modelos de seguros como se

observa en la figura 6.12, vemos que estos modelos de renta son muy sensibles a ésta característica.

Si para una tasa de interés del 3% comparamos los valores actuariales, de los modelos MSC-1 y MRD-1 (los cuales tienen asignados la misma tabla de mortalidad, también MSC-1 tiene indemnización constante y MRD-1 tiene pagos constantes) tabulados en las tablas XII, XIX respectivamente, tenemos que para MSC-1 es 0.5727 mientras que para MRD-1 es 25.8987, vemos una diferencia considerable, esto se debe principalmente a la naturaleza misma de los modelos, mientras los modelos de seguros pagan si el ente f/q en un intervalo de tiempo, los modelos de rentas pagan si el ente sobrevive a dicho intervalo.

Tabla XX

Valor Actuarial para los modelos MRD-3, MRD-4, variando la tasa de interés

Tasa de Interés	MRD-3	MRD-4
0,03	479,42764	463,831
0,04	370,87266	360,07145
0,05	291,88222	284,3001
0,06	233,56969	228,17092
0,07	189,89892	185,99683
0,08	156,72445	153,86
0,09	131,16939	129,03258
0,1	111,21435	109,59413
0,11	95,4266	94,17781
0,12	82,77807	81,79984
0,13	72,52292	71,74428
0,14	64,11397	63,4846
0,15	57,14535	56,62918
0,16	51,31282	50,88365
0,17	46,38585	46,02441
0,18	42,18792	41,87995
0,19	38,58258	38,31731

La tabla XX fue calculada considerando los modelos con pagos incrementales MRD-3, MRD-4 se observa que aunque tienen distintos tipos de pagos estos modelos también siguen el patrón decreciente que tienen sus homólogos MRD-1 y MRD-2, al afrontarlos a variaciones en la tasa de interés.

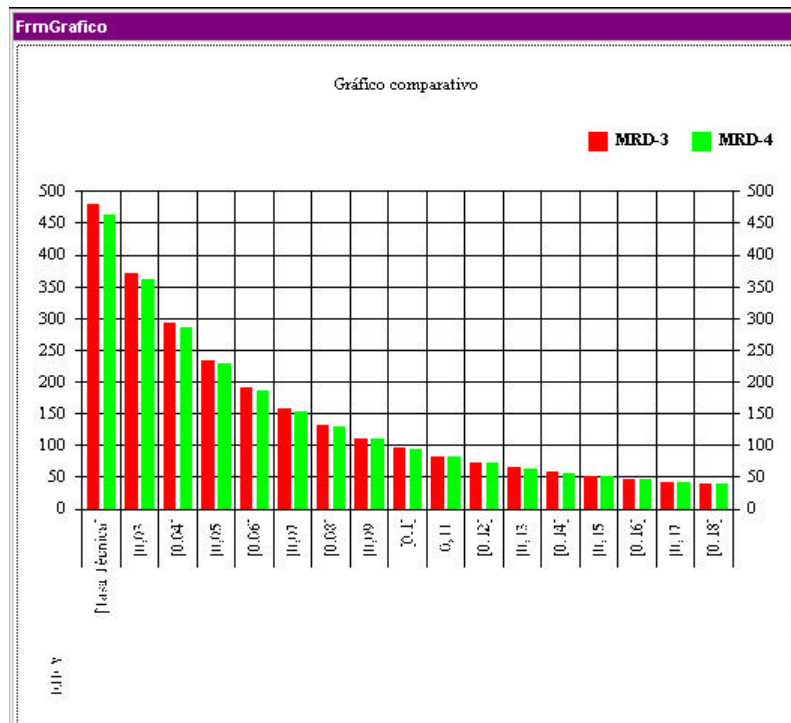


Figura 6.21 Valores Actuariales de los modelos MRD-3, MRD-4, variando la tasa de interés

La figura 6.21 muestra el patrón de decrecimiento del valor actuarial de los modelos incrementales, la figura 6.22 y 6.21 nos conduce a concluir que la tasa de interés sensibiliza en gran medida los modelos de renta, a tal punto que les causa a sus valores actuariales un efecto decreciente cóncavo hacia abajo.

6.5.5.2. Modelos de rentas sensibilizados a través de la edad del asegurado

Tabla XXI

Valores actuariales para modelos de renta discretos, variando la edad del asegurado.

Edad	MRD - 1	MRD - 2
20	25,38987	24,94094
25	24,10809	23,68643
30	22,63048	22,19979
35	20,9333	20,47481
40	18,9987	18,51467
45	16,82411	16,32049
50	14,36581	13,87052
55	11,57036	11,13533
60	8,35404	8,05391
65	4,57314	4,46897
70	0	0

Los valores actuariales de la tabla XXI fueron obtenidos por medio de los modelos MRD-1, MRD-2, variando la edad del asegurado, fijada inicialmente en 20 años de edad hasta llegar a la vigencia de los modelos fijada en 70 años de edad.

Los valores actuariales decrecen conforme aumenta la edad del asegurado como se aprecia en la tabla XXI, además se observa

que la diferencia entre los modelos actuariales de MRD-1 y MRD-2 es mínima.

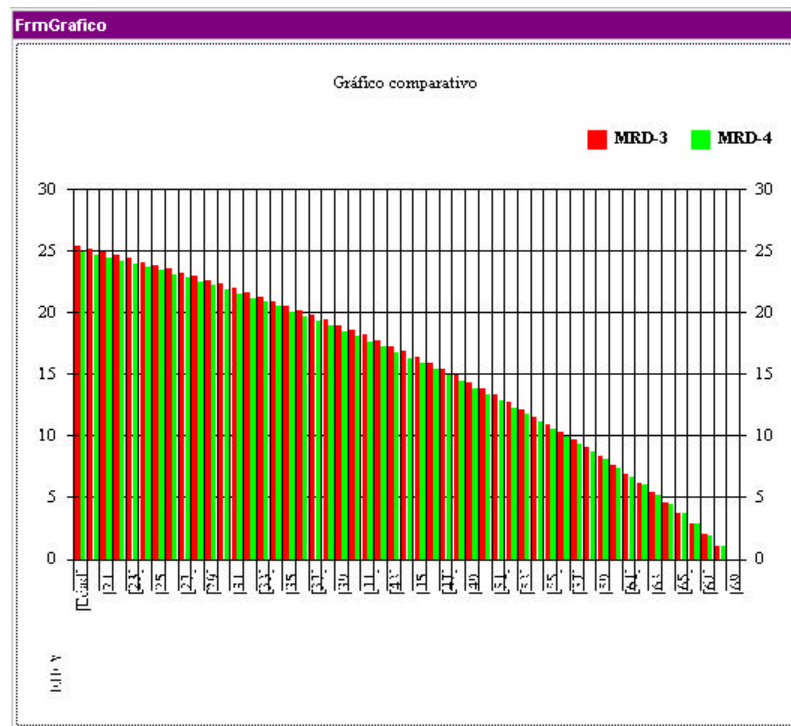


Figura 6.22 Gráfico de los valores Actuariales para los modelos MRD-1 , MRD-3, variando la edad del asegurado.

Algo interesante que se obtiene al comparar los gráficos 6.22 y 6.20 es que ambos presentan un decrecimiento en su valor actuarial, pero la tasa de interés causa un efecto de decrecimiento más efectivo que la edad del asegurado.

6.5.5.3. Modelos de rentas sensibilizados a través de los años de diferimiento

Tabla XXII

Valor actuarial para los modelos de rentas, variando los años de diferimiento

Diferimiento	MRD - 2	MRD - 1
20	24,94094	25,38987
25	20,24118	20,68255
30	16,22379	16,64505
35	12,7884	13,18497
40	9,85499	10,22405
45	7,36014	7,69854
50	5,25309	5,55591
55	3,49237	3,75035
60	2,04549	2,24362
65	0,88769	1,00161
70	0	0

La tabla XXII se generó variando los años de diferimiento de los modelos MRD-2 y MRD-1 los cuales inicialmente tenían cero años de diferimiento. Se observa que el valor actuarial de los modelos disminuye a medida que aumentan los años de diferimiento.

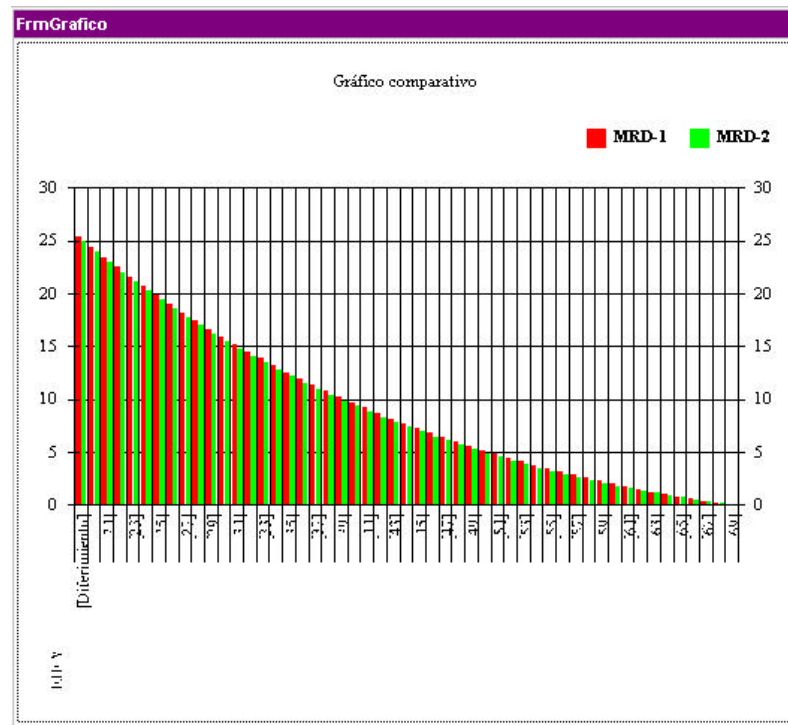


Figura 6.23 Gráfico de los valores Actuariales para los modelos MRD-1, MRD-2 variando los años de diferimiento.

La figura 6.23 se puede observar la forma en que decrece el valor actuarial al aumentar los años de diferimiento, el patrón de decrecimiento sigue conservándose, pero al comparar las figuras 6.23, 6.22 y 6.20 se observa que estos decrecen de formas diferentes.

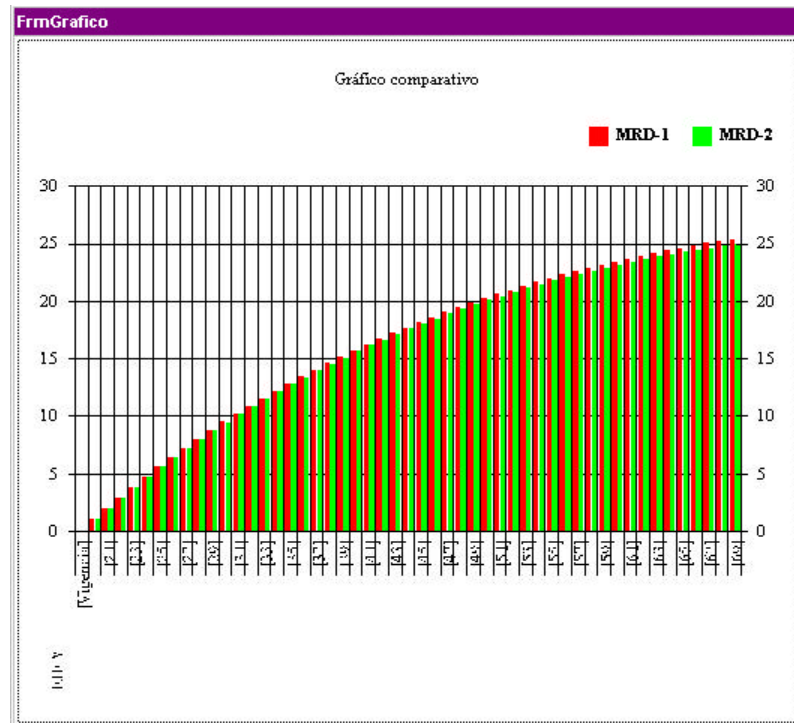
6.5.5.4. Modelos actuariales sensibilizados a través de la vigencia del modelo

Tabla XXIII

Valor actuarial para los modelos de rentas discretos variando la vigencia de los modelos

Vigencia	MRD - 1	MRD - 2
20	0	0
25	4,70732	4,69976
30	8,74482	8,71715
35	12,2049	12,15254
40	15,16582	15,08596
45	17,69133	17,5808
50	19,83396	19,68785
55	21,63952	21,44857
60	23,14625	22,89545
65	24,38827	24,05326
70	25,38987	24,94094

La tabla XX fue generada a partir de los modelos MRD-1, MRD-2, se procedió a variar el año de vigencia inicialmente fijado en los 70 años de edad. Esta tabla nos deja ver el carácter creciente del valor actuarial.



**Figura 6.24 Valor Actuarial para los modelos MRD-1, MRD-2
variando el año de vigencia**

La figura deja ver la forma en que crece el valor actuarial de los modelos considerados al variar la vigencia, observamos que esta característica causa un gran efecto en los modelos considerados.

6.6. Módulo para el análisis Multi – modelos

Este módulo considera los modelos de seguros y rentas discretos, generados por las bibliotecas de vínculos dinámicos (DLL), para así generar los planes de obligaciones y los planes de pagos, formando un plan general. A éste plan general Maxim da la posibilidad de obtener la prima neta, de inventario y comercial, así como las respectivas reservas matemáticas.

6.6.1. Inicio de un plan

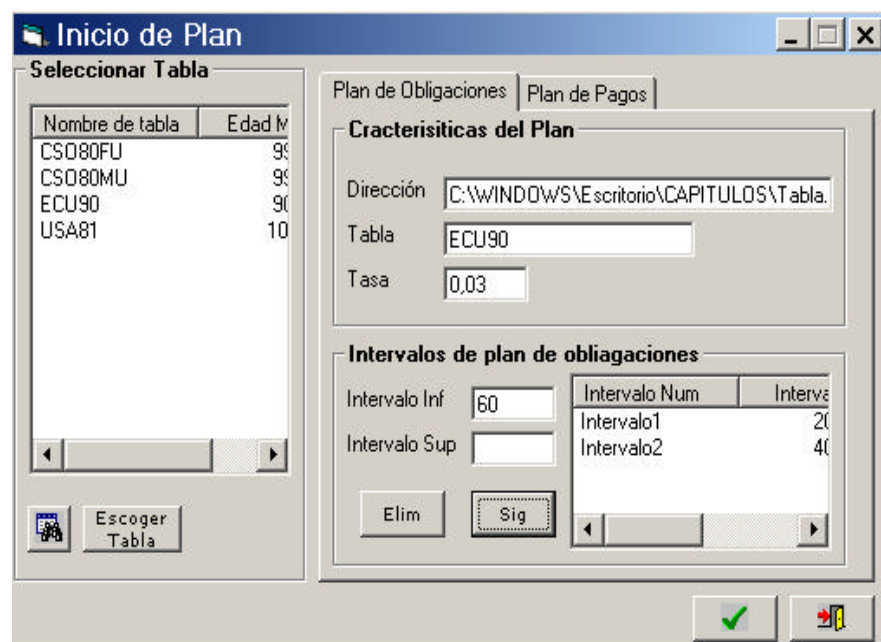


Figura 6.25 Interfaz para el Inicio de un plan

La figura 6.25 muestra la interfaz visual para iniciar un plan, en esta interfaz el usuario va a asignar una tabla de mortalidad al plan, así como la tasa de interés. También va a definir los intervalos de edades para el plan de obligaciones y el plan de pagos, en cada intervalo se va a asignar un modelo actuarial como se verá posteriormente.

Para este ejemplo particular se va a fijar las siguientes características iniciales para un plan.

Tabla XXIV

Características iniciales para plan a generar

Tabla	ECU90	
Tasa de Interés	3%	
Intervalos Plan Obligaciones	[20-40)	[40-60)
Intervalo Plan de Pagos	[20-60)	

La Tabla XXIV muestra las condiciones iniciales del plan a generar, se aprecia que el plan de obligaciones se ha particionado en dos intervalos, mientras que el plan de pagos tiene un intervalo, esto quiere decir que el plan de obligaciones estará compuesto de dos modelos actuariales, uno por cada intervalo, no

así el plan de pagos, éste sólo puede estar compuesto de un modelo actuarial.

6.6.2. Generar plan de obligaciones

Plan de obligaciones

Seguro | Renta

Intervalo

Con modelo Sin modelo **Intervalo 1 : [20 , 40]**

Seguros

Tipo de Indemnización: Constante Discreto C

Periodo para cancelar: Anual

Momento de pago: Al finalizar el periodo de f/q

Generar

Modelos del Plan de Obligaciones SigMod

Sub Intervalo	Codigo	Clase de Modelo
Intervalo 1 : [20 , 40]	MSD - 1	Seguro

Figura 6.26 Interfaz para el plan de obligaciones

La figura 6.26 muestra la interfaz que permite asignar los modelos actuariales a los intervalos definidos para el plan de obligaciones, se observa que existe la posibilidad de asignar a un intervalo un

modelo de seguros o un modelo de renta, la tabla siguiente muestra los modelos establecidos para cada intervalo.

Tabla XXV

Modelos para plan de obligaciones

Intervalo	Código del modelo	Descripción
[20-40)	MSD-1	Modelo discreto de seguro con indemnización constante para una persona de 20 años con vigencia hasta los 40 años
[40-60)	MSD-2	Modelo de seguro con indemnización incremental para una persona de 20 años diferido hasta los 40 años, vigente hasta los 60 años.

La tabla XXV muestra los modelos asignados a cada intervalo, cada uno representado por un código, por la tabla XXIV se deduce que los modelos presentados tienen la tabla de mortalidad ECU90, y una tasa de interés del 3%.

6.6.3. Generar plan de beneficios

Intervalo
 Con modelo Sin modelo Intevalo 1 : [20 , 60]

Rentas

Tipo de Pago: Constante [C]
 Periodo de Pago: Anual
 Pago De Cuota: Anticipado

Generar

Modelos del Plan de Pagos Sig

Sub Intervalo	Codigo	Edad	Diferin
Intevalo 1 : [20 , 60]	MRD - 1	20	0

Figura 6.27 Interfaz visual para el plan de pagos

Esta interfaz realiza la misma función que la interfaz presentada en la figura 6.26, esto es generar valores actuariales y asignarlos a un intervalo, pero para el plan de pagos o beneficios. En este nivel sólo se pueden generar modelos de rentas.

Tabla XXVI

Modelos asignados al plan de pagos

Intervalo	Código del modelo	Descripción
[20-60)	MRD-1	Modelo de renta discreto con pago constante anticipado, para un ente de 20 años con vigencia hasta los 60 años de edad

La tabla XXVI muestra el modelo establecido en el único intervalo definido en el plan de pagos, éste modelo se lo identifica por su respectivo código.

6.6.4. Interfaz para el calculo de primas y reservas de un plan

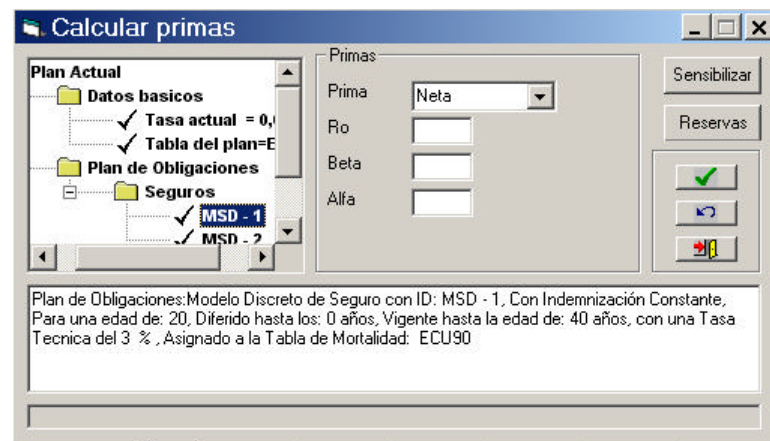


Figura 6.28 Interfaz para calcular primas y reservas

La figura 6.28 muestra la interfaz visual, para el cálculo de los diferentes tipos de primas y reservas, al llegar a esta altura ya se tienen que haber definido los planes de obligaciones y de pagos, como se puede apreciar en el árbol de la parte izquierda de la figura, éste árbol muestra de una manera jerárquica y organizada las características del plan generado.

El árbol que representa las características del plan posee tres carpetas superiores, los cuáles son “datos básicos del plan” donde se detalla la tasa de interés y la tabla de mortalidad asignada al plan, en la segunda carpeta “Plan de obligaciones” se encuentran representados a través de su código todos los modelos actuariales que componen el plan de obligaciones, así mismo la tercera carpeta “Plan de pagos” tenemos todos los modelos actuariales de rentas que componen al plan de pagos.

En la parte inferior de la pantalla se observa el detalle de un modelo actuarial seleccionado del árbol, este árbol es de gran importancia para el usuario ya que ayuda a recordar los detalles del plan generado.

6.6.4.1. Prima neta del plan y reserva matemática de la prima

Tabla XXVII

Prima neta para distintas tasas de interés

Tasa del plan	Valor Actuarial
0,03	0,02402
0,04	0,02009
0,05	0,01673
0,06	0,0139
0,07	0,01153
0,06	0,00958
0,09	0,00797
0,1	0,00666
0,11	0,00559
0,12	0,00473
0,13	0,00403
0,14	0,00346
0,15	0,00301
0,16	0,00264
0,17	0,00235
0,18	0,00211
0,19	0,00191
0,2	0,00175

La tabla XXVII muestra los resultados de calcular la prima neta al plan generado, para distintas tasas de interés, recordemos que el plan generado ésta formado por tres modelos actuariales, dos modelos de seguro para el plan de obligaciones, y un modelo de renta para el plan de pagos.

Se observa el patrón decreciente de la prima neta al exponerlo a variaciones de la tasa de interés del 3% hasta el 20%.

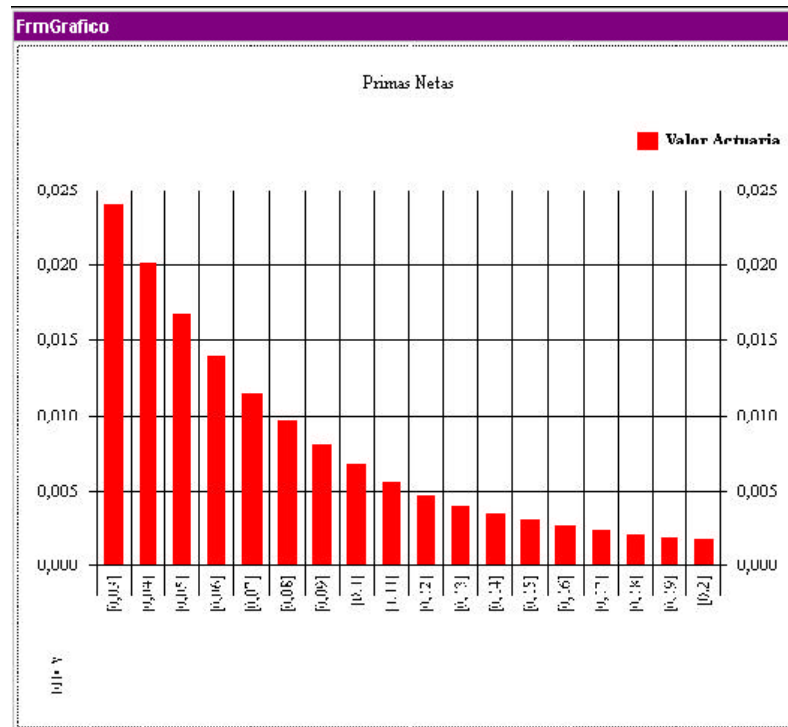


Figura 6.29 Gráfico de la prima neta expuesto a diferentes tasas de interés

La figura 6.29 deja ver que la tasa de interés al igual que en los modelos de seguros y rentas sensibiliza de gran forma la prima neta, también de una manera cóncava hacia abajo.

Tabla XXVIII

**Reservas matemáticas para la prima neta para distintas tasas
de interés**

Edad \ Tasa	Reserva matemática		
	3%	5%	10%
20	0	0	0
25	0,1250146	0,0903172	0,0370131
30	0,2704926	0,2059151	0,0965412
35	0,4384618	0,3523102	0,1902268
40	0,6335413	0,5390277	0,3396674
45	0,8169294	0,7332858	0,5325971
50	0,8926553	0,8395349	0,6881026
55	0,7111421	0,6970249	0,6363182
60	0	0	0

La tabla XXVIII muestra las reservas matemáticas para distintas tasas de interés y edades, para cada tasa existe un patrón decreciente hasta cierta edad cuando comienza a crecer nuevamente, para este caso en particular ésta edad está en los 50 años, entonces la reserva para este plan tiene un máximo en los 50 años, como se apreciara en la siguiente figura.

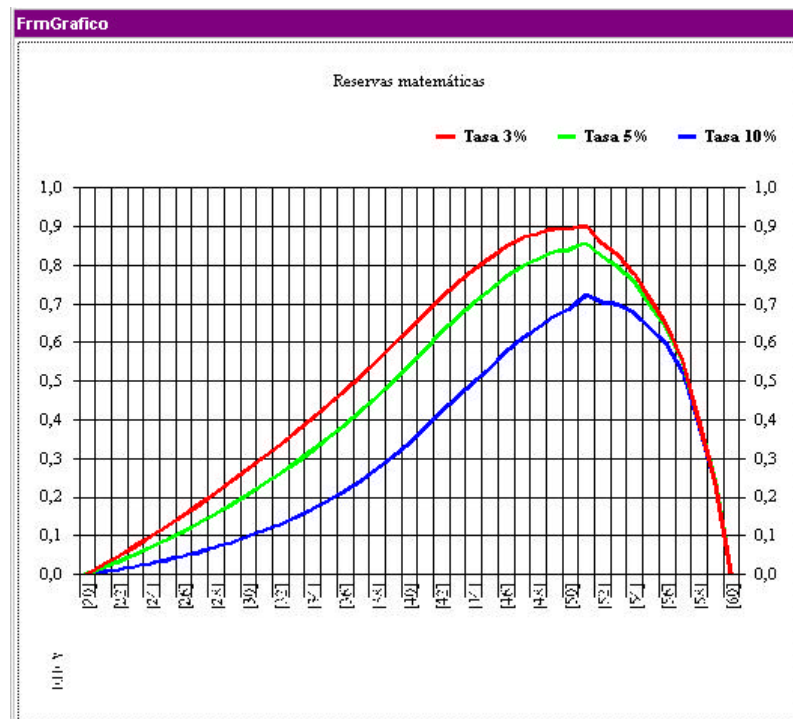


Figura 6.30 Reserva matemática para la prima neta a distintas tasas de interés.

La figura 6.30 deja ver claramente el máximo común que presenta la reserva para todas las tasas de interés, además mientras la tasa de interés disminuye la reserva crece más decisivamente.

Un detalle interesante que se observa en la figura 6.30 es que existe un punto en común para las tres funciones de reservas donde las reservas decrecen de igual forma, es decir las reservas

son inmunes a la variación de la tasa de interés, esto ocurre a partir de los 55 años de edad.

6.6.4.2. Prima de inventario y reserva matemática de la prima

Tabla XXIX

Primas de Inventario calculadas con gastos de administración igual al 10 % de la suma asegurada, para distintas tasas de interés

Tasa del plan	Valor Actuarial
0,03	0,02402
0,04	0,02009
0,05	0,01673
0,06	0,0139
0,07	0,01153
0,08	0,00958
0,09	0,00797
0,1	0,00666
0,11	0,00559
0,12	0,00473
0,13	0,00403
0,14	0,00346
0,15	0,00301
0,16	0,00264
0,17	0,00235
0,18	0,00211
0,19	0,00191

La tabla XXIX muestra los valores de las primas de inventario para el plan generado, observamos un patrón decreciente al igual que la

prima neta, esta característica también sensibiliza considerablemente la prima de inventario, los gastos de administración considerado para este ejemplo fue del 10% de la suma asegurada.

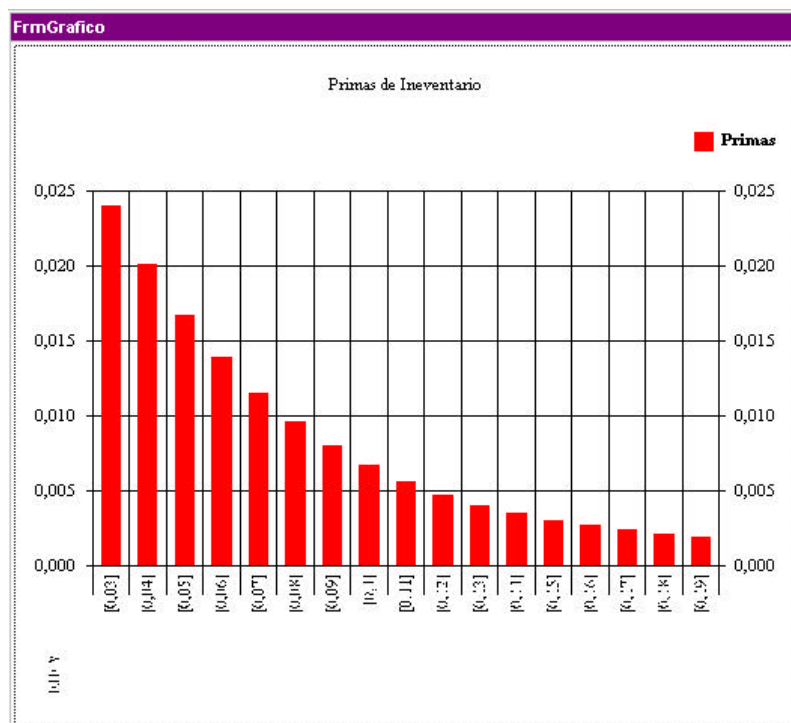


Figura 6.31 Gráfico para las primas de inventario para distintas tasas de interés.

Al igual que las primas netas, el patrón de decrecimiento es cóncavo hacia abajo, pero a diferencia de las primas netas, las primas de inventario presentan un valor más alto, esto se debe

principalmente al porcentaje de los gastos administrativos que consideran las primas de inventario.

Tabla XXX

Reserva matemática para la prima de inventario, para diferentes tasas de interés

Edad\Tasa	3%	5%	10%
20	0	0	0
25	0,1250146	0,0903172	0,0370131
30	0,2704926	0,2059151	0,0965412
35	0,4384618	0,3523102	0,1902268
40	0,6335413	0,5390277	0,3396674
45	0,8169294	0,7332858	0,5325971
50	0,8926553	0,8395349	0,6881026
55	0,7111421	0,6970249	0,6363182
60	0	0	0

La tabla XXXI fue generada obteniendo las reservas matemáticas de la prima de inventario para distintas tasas de interés y edades, observamos la tendencia creciente hasta los 50 años de edad, donde se presenta el máximo para las tres series, después de este valor la reserva desciende hasta llegar a una reserva cero a los 60 años, es decir hasta la llegar a la vigencia de los modelos.

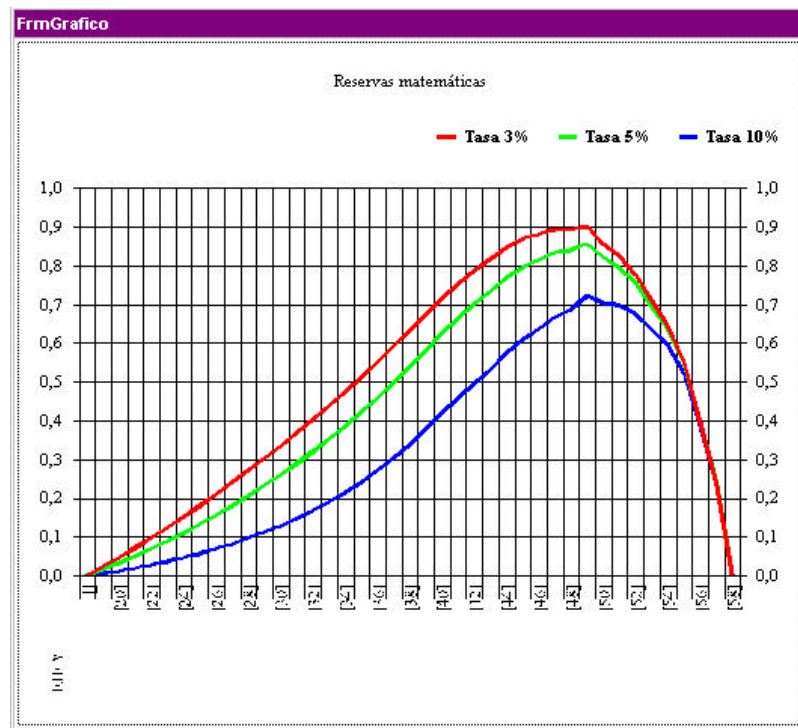


Figura 6.32 Reservas matemáticas para distintas primas de inventario para tasas de interés diferentes.

Al observar la figura 6.32 se aprecia que la tasa de interés sensibiliza la reserva de tal forma que a menor tasa menor mayor reserva matemática.

Al comparar las figuras 6.31 y 6.32 se observa que las reservas en 6.32 presentan máximos a edades más tempranas, además las reservas calculadas con las primas de inventario son mayores a

las reservas calculadas con las primas netas, para todas las edades.

6.6.4.3. Prima comercial y reserva matemática para la prima

Tabla XXXI

Primas comerciales para diferentes tasas de interés.

Tasa	Prima Comercial
0.03	0.04218
0.04	0.03972
0.05	0.03801
0.06	0.03698
0.07	0.03654
0.08	0.0366
0.09	0.03709
0.1	0.03793
0.11	0.03905
0.12	0.04039
0.13	0.04192
0.14	0.04358
0.15	0.04534
0.16	0.04718
0.17	0.04907
0.18	0.051
0.19	0.05295
0.2	0.05491

La tabla XXXI fue obtenida sobre la base del plan generado anteriormente, la prima comercial fue calculada con un gasto adquisición inicial del 30% de la suma asegurada (comisiones

para los agentes y corredores), seguidos del 3% de los gastos de cobranza (cobradores domiciliarios o cobros electrónicos) aplicado a la prima comercial Para cada año de pago de la prima comercial y de 0.1% de los gastos de administración (salarios, alquiler de oficina) aplicado a la suma asegurada para cada año de vigencia de la póliza.

La tabla XXXI muestra que la tasa de interés causa un efecto diferente a las primas comerciales, puesto que éstas no decrecen para todo el rango de tasas, si no que vuelven a crecer al llegar a un 8% de interés.



Figura 6.33 Primas comerciales, para dis tintas tasa de Interés

La figura 6.33 deja ver el comportamiento de la prima comercial, al someterla a una variación de un punto porcentual en la tasa de Interés, se observa que inicialmente la prima comercial decrece hasta llegar al 8%.

Tabla XXXII

Reserva matemática para la prima comercial, para diferentes tasas y edades.

Edad Tasa	3%	5%	10%
20	0	0	0
25	-0.2355125	-0.2456928	-0.2766735
30	-0.0575674	-0.1085239	-0.2088966
35	0.1473948	0.0646748	-0.1028366
40	0.3848593	0.2849697	0.0655994
45	0.6166651	0.5211728	0.2877294
50	0.748644	0.6809047	0.4897183
55	0.6327604	0.6069382	0.5126097
60	0	0	0

Las reservas matemáticas presentadas en el cuadro XXXII muestra en los primeros años reservas negativas esto nos indica que en estos años los ingresos de las primas por pagar son mayores a los egresos por pagos futuros de indemnización, a partir de los 31 años para la prima calculada con una tasa de Interés de 3%, se presentan reservas positivas, este mismo caso ocurre para la

prima calculada a una tasa del 5% a los 33 años, mientras que para la prima calculada al 10% a los 38 años.

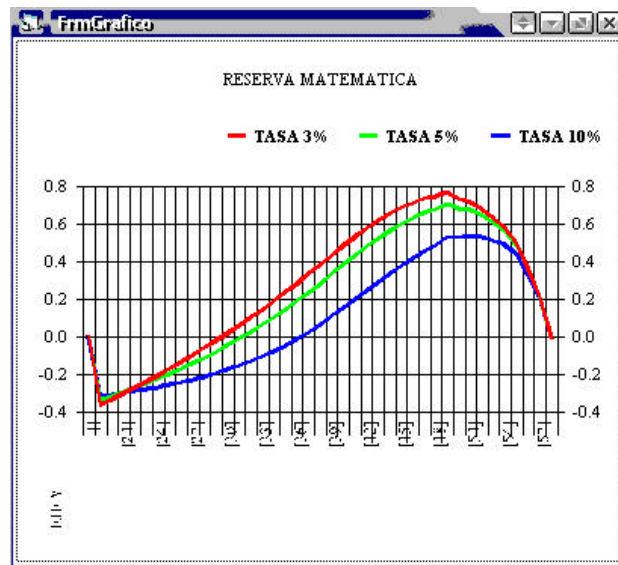


Figura 6.34 Reserva matemáticas de las primas comerciales, para diferentes tasas de interés.

La figura 6.34 muestra las reservas matemáticas de las primas comerciales para tasas del 3,5 y 10 % de interés, observamos que la diferencia con las figuras 6.30 y 6.32 no es significativa en el patrón de crecimiento y decrecimiento, más bien la diferencia principal radica que en la figura 6.34 las reservas son negativas al inicio del periodo.

Se observa que al igual que las reservas obtenidas de las primas de inventario y primas netas, las primas comerciales también presentan reservas comunes para las tres tasas de interés, a partir de los 50 años de edad.

RESUMEN

El presente trabajo presenta el diseño desarrollo e implementación de una hoja electrónica actuarial para el análisis de los modelos de seguros y rentas, a través de sus valores actuariales, así como el cálculo de primas netas de inventario y comercial con sus reservas respectivas para distintos planes actuariales, desarrollado bajo el lenguaje de programación Visual Basic 6.0 incluidas las herramientas que este presenta para la creación de componentes de código Active X, el nombre del proyecto es **“Diseño e implementación de una hoja electrónica para el análisis actuarial”**.

El primer capítulo menciona la evolución de las hojas electrónicas a través del tiempo, así cómo la carencia en la flexibilidad para el análisis de los modelos actuariales, además se revisan algunas bondades de la hoja electrónica de Microsoft Excel.

En la segunda parte se da una breve introducción a la orientación a objetos y consecuentemente a la programación orientada a objetos, mencionando los conceptos más importantes que componen este nuevo paradigma de programación.

El tercer capítulo aborda en primera instancia las bases teórica de las matemáticas actuariales, para después considerar la mayor parte de los modelos de seguros y rentas existentes, así como las primas y reservas matemáticas que se forman con estos modelos.

El cuarto capítulo trata sobre las herramientas informáticas utilizadas para desarrollar la hoja electrónica.

El quinto capítulo desarrolla de manera detallada el diseño de la hoja electrónica, desde el análisis de las necesidades del sistema, hasta el diseño de las DLL Active X para seguros y rentas, así como de la interfaz visual.

El sexto capítulo muestra la hoja electrónica en tiempo de ejecución, se brindan los resultados que proporciona la hoja electrónica y el consecuente análisis de los resultados sobre la base de algunos modelos de seguros y rentas generados por el sistema.

INTRODUCCION

MAXIM es el nombre del proyecto “Hoja electrónica actuarial”, el mismo que está compuesto por un ejecutable llamado Maxim.exe, y dos componentes de código DLL Active X, el uno llamado Seguros.dll, y el otro Rentas.dll.

La aplicación ejecutable proporciona la interfaz visual de una hoja electrónica, así como las herramientas de edición, además está compuesta de dos módulos principales, uno para el análisis unimodelos y otro para el análisis multimodelos o planes, esta aplicación sirve de cliente para las DLL Active X.

Los componentes de código Seguros.dll y Rentas.dll son servidores de los modelos actuariales, estos componentes al admitir la automatización Active X pueden ser utilizados por aplicaciones diferentes a Maxim, como Microsoft Excel.

Maxim permite sensibilizar los modelos actuariales o primas individualmente o en grupo y analizar el impacto que esto tiene en sus valores actuariales, y reservas matemáticas respectivamente.

CONCLUSIONES Y RECOMENDACIONES

CONCLUSIONES

1. Maxim es un sistema integrado por tres proyectos dependientes Maxim.exe, y los componentes de código Seguros.dll y Rentas.dll, siendo estos dos últimos proyectos no exclusivos para Maxim, si no que se los puede utilizar en aplicaciones que admitan automatización Active X.
2. Los objetos son una herramienta poderosa para la programación, los cuales permiten representar y organizar los modelos actuariales de una mejor manera, al tener estas características exclusivas que los distinguen unos de otros.
3. La programación orientada a objetos facilita el diseño de los modelos actuariales, debido que permite conceptualizar de una mejor forma los objetos intangibles pudiendo así generar cualquier tipo de modelo seguros y rentas, Los modelos actuariales fueron bien representados por los objetos que se generaron en los componentes de código DLL Active X.

4. La expansibilidad de Seguros.dll, y Rentas.dll es una tarea sencilla debido a la forma en que fueron estructurados los componentes, es decir que en una nueva versión de los componentes de código es posible añadir más modelos con mínimos cambios en la codificación.
5. Los componentes visuales de Visual Basic pueden ajustarse a las necesidades del programador, al tratarlos como variables de objeto, de ésta manera se los puede organizar en módulos de clase y personalizar a conveniencia nuevas propiedades y métodos.
6. Al tratar las tablas de mortalidad como características de los modelos discretos, y las leyes de mortalidad como características para los modelos continuos se enriqueció el poder de análisis de Maxim, ya que la sensibilidad de modelos no se limita a las variaciones de características como la tasa técnica o la edad del asegurado, si no también a variaciones de tablas o leyes.
7. El módulo 2 de Maxim, aparte de ser didáctico es el más comercial de los dos que componen el proyecto, puesto que brinda la posibilidad de analizar actuarialmente planes, los cuales son los

que se ofrecen en las compañías de Seguros con el nombre de productos.

8. Basándose en el análisis de los resultados obtenidos en el capítulo 6 se puede concluir que la tasa de interés sensibiliza en gran medida los valores actuariales de los modelos de seguros y rentas de modo que a mayor tasa de interés menor valor actuarial.
9. El riesgo para los modelos constantes de seguros y rentas también son afectadas por la variación en la tasa de interés de manera decreciente, pero este decrecimiento se presenta de manera menos pronunciada que en los valores actuariales para los mismos modelos.
10. Las variaciones en la edad del asegurado también afectan al valor actuarial de los modelos con indemnización constante, de manera creciente hasta una cierta edad, desde ahí comienza a decrecer significativamente hasta llegar a un valor actuarial de cero al llegar a la edad de vigencia del modelo.
11. Para los modelos con indemnización incremental la variación de la edad del asegurado causa un efecto de decrecimiento en el valor

actuarial, mientras la edad del asegurado se acerca a la vigencia del modelo.

12. El modelo de dote puro incrementa su valor actuarial mientras la edad del asegurado se aproxima a la vigencia del modelo hasta llegar a la indemnización total del modelo, que para estos casos era de 1 u.m.
13. La variación en el diferimiento causa el mismo efecto decreciente en el valor actuarial, esto es a mayor años de diferimiento de los modelos constantes e incrementales, menor valor actuarial.
14. De las tres características consideradas las tasa de interés la edad del asegurado, y el diferimiento de los modelos, el que sensibiliza en mayor medida es la tasa técnica.
15. Para los modelos constantes e incrementales se cumple que mientras mayor sea la edad de vigencia de los modelos mayor es el valor actuarial.
16. Las primas netas presentan un valor actuarial menor que las primas de inventario, y estas menor a las primas comerciales.

17. Las primas netas, de inventario y comerciales compuestas por modelos constantes e incrementales decrecen al aumentar la tasa de interés.
18. Las reservas matemáticas para las primas netas, de inventario disminuyen conforme aumenta la tasa de interés.
19. Las reservas matemáticas calculadas para un prima en particular a diferentes tasas de interés presentan un máximo en común.
20. Las reservas matemáticas calculadas para una prima en particular presentan reservas comunes a partir de una edad determinada, es decir que a partir de esa edad las reservas no son sensibles a las variaciones en la tasa técnica.

RECOMENDACIONES

1. El ICM por medio de la oficina de promoción y servicios estadísticos puede ser la entidad pionera en el desarrollo del análisis de la realidad nacional actuarial, guiando la investigación para obtener tablas de mortalidad diversificadas por sexo, provincias, estratos, etc. Y no solo para el evento de muerte, si no también de despido, de accidentes laborales, para que los productos que se elaboren estén acordes a nuestra realidad, y no a la realidad de ciudades de países vecinos como Colombia, o España.
2. Se recomienda como un trabajo posterior a la investigación, el diseño y desarrollo de un sistema que se integre a Maxim, donde se procesen los datos brutos, de defunciones, que permita de una manera técnica la creación de tablas de mortalidad, así como también un módulo que permita ajustar una ley de mortalidad teórica a una tasa de mortalidad.
3. También con este trabajo se recomienda que los profesionales que trabajen en el campo actuarial se actualicen en el uso de herramientas computacionales actuales, como los componentes y

controles Active X, con los que se desarrollará aplicaciones más robustas y profesionales en esta área.

4. Maxim debe de ser un punto de partida para los proyectos de software de la materia matemáticas actuariales, para que el objetivo de estos proyectos no sea solo el cálculo de primas, valores actuariales, o reservas si no el análisis de modelos e interpretación de resultados.

BIBLIOGRAFIA

1. J.F. Lawless, Statistical models and methods for lifetime data. Editorial Wiley.
2. Swiss Re Life & Health, seminario de actuariales, noviembre del 2001
3. Newton L. Bowers, JR, Hans U. Gerber, James C. Hickman, Donald A. Jones, Cecil J. Nesbitt, Actuarial mathematics, Editado por Millicent M. Treloar 1986.
4. Gerald and Wheathley, Applied numerical analysis sexta edición, Edición Addison – Wesley
5. Keneth E. Kendall, Julie E. Kendall, Análisis y diseño de sistemas tercera edición. Editorial Prentice Hall.
6. James Rombaugh, Michael Bluha, William Premerluni, Frederick Eddy, William Lorensen, Modelo y diseño orientado a objetos. Editorial Prentice Hall

7. James Martin, James J Odell, Métodos orientados a objetos. Editorial Prentice Hall.
8. Gefrey Whitten, Victor M. Borlow, Análisis y diseño de sistemas de información tercera edición. Edición Mac Graw Hill.
9. Dan Appleman Desarrollo de componentes COM / Active X con Visual Basic 6, Editorial Prentice Hall 2000.
10. Francesco Balena Programming Mocosoft Visual Basic 6.0.