Documento de Especificación de Requisitos de Software (SRD)

para

Simulador de expansión de virus

Versión 1.0

Sergio Flor García 48904986V

Introducción a la Ingeniería de Software.   
Prácticas curso 2019/20

6 de abril de 2020

Tabla de Contenidos

1. Introducción 1

1.1 Objetivo 1

1.2 Ámbito 1

1.3 Definiciones, siglas y abreviaturas 1

1.4 Referencias 2

1.5 Panorámica del documento 2

2. Descripción General 2

2.1 Relación con otros proyectos 2

2.2 Relación con proyectos anteriores y posteriores 2

2.3 Objetivos y funciones 2

2.4 Consideraciones del entorno 3

2.5 Relaciones con otros sistemas 3

2.6 Restricciones generales 3

2.7 Descripción del modelo. 3

3. Requisitos Específicos 7

3.1 Requisitos funcionales 7

Apéndice único: Enunciado de la práctica 10

Historial de Revisiones

|  |  |  |  |
| --- | --- | --- | --- |
| **Nombre** | **Fecha** | **Razón de los cambios** | **Versión** |
|  |  |  |  |
|  |  |  |  |

# Introducción

## Objetivo

El objetivo de esta práctica es la consolidación de los conocimientos adquiridos con el estudio de la asignatura mediante un ejercicio que abarca los contenidos del curso, en este caso la realización de un programa orientado a objetos en Java que simule la expansión de un virus en una población siguiendo un modelo exponencial

Un objetivo secundario es comparar como se aborda la realización de un proyecto software desde la óptica de la ingeniería frente al ya conocido y realizado desde la óptica del desarrollador o el programador.

Se establece el siguiente calendario de fechas de entrega para las distintas partes de las prácticas:

* Práctica 1 - Diseño: 19 de abril.
* Práctica 2 - Codificación: 10 de mayo.

## Ámbito

El ámbito del proyecto se establece en la formalización del proceso de desarrollo de una aplicación partiendo de su especificación. **Aunque se señala a la descripción pormenorizada de la aplicación ―apéndice A del enunciado de las prácticas― como punto de partida, parece conveniente redactar el documento de especificación de requisitos para aplicar las herramientas y estructura correspondientes a esta parte del proceso de desarrollo.**

Como es sabido, esta formalización del proceso de desarrollo se realizará en dos partes:

1. Elaboración del documento de diseño del proyecto.
2. Codificación de la aplicación siguiendo los documentos de diseño que se entreguen en la práctica 1.

## Definiciones, siglas y abreviaturas

* **E** : Número de contactos que en promedio tenga cada infectado con personas no infectadas.
* **p** : Probabilidad de infectarse de un contacto.
* **V** : Número de viajeros diarios de una comunidad a otra.
* **N** : Número de infectados de una comunidad que distingue entre:
  + **Nd** : número de infectados un día concreto.
  + **Nd+1 :** Número de infectados al día siguiente.
  + **Nv :** Número de infectados por culpa de los viajeros.

## Referencias

1. Enunciado de las prácticas curso 2019-2020 para la asignatura Introducción a la Ingeniería de Software.
2. Gómez Palomo, S., & Moraleda Gil, E. (2020). *Aproximación a la Ingeniería de Software* (segunda ed.). Madrid: Ramon Areces.

## Panorámica del documento

El resto del documento recoge la descripción del modelo del sistema y la lista de requisitos específicos utilizando el método de Análisis Estructurado y complementado con un estudio de Casos de Uso. Para todo ello se utilizan e incluyen los diagramas de flujos de datos, los de entidad-relación, el diccionario de datos y los casos de uso del sistema.

# Descripción General

## Relación con otros proyectos

Ninguna

## Relación con proyectos anteriores y posteriores

Ninguna.

## Objetivos y funciones

El sistema debe simular la expansión de un virus dentro de la población de una comunidad y entre las poblaciones de distintas comunidades a partir de un primer infectado en una de estas comunidades. Para ello el sistema debe calcular:

* El número de infectados de un día para otro según la fórmula:

**Nd+1** = Nd + Nd \* E \* p = **Nd \* (1 + E \* p)**

* El número de infectados en una comunidad por culpa de los viajeros ― Nv ― con la fórmula:

**Nv**= **E \* p \* V \* ( Nd,origen/Porigen )**

Donde Nd,origen sería número de infectados en la comunidad de origen del viajero y Porigen sería la población de la comunidad de origen.

* El total de infectados en una comunidad sumando el resultado de los cálculos anteriores.

Además el sistema debe permitir introducir al usuario los siguientes datos:

* Número de comunidades y población de cada una de ellas.
* Porcentaje de habitantes en cada comunidad que viaja diariamente a cada una de las otras comunidades, igual para todas las comunidades para simplificar.
* Coeficientes E y p.(Aunque el enunciado incluye V, el dato no es un coeficiente sino que se describe como el número de viajeros y por tanto se calcula en base al porcentaje del punto anterior)
* Número de días a simular.

Estos datos deben poder introducirse por primera vez, y una vez terminada la simulación, deben poder modificarse para repetirla.

Como resultado, el programa debe devolver una tabla con los datos diarios de:

* Infectados cada comunidad.
* Porcentaje de población infectada en cada comunidad.
* Infectados de la población total.
* Porcentaje de población total infectada.

Además, la presentación de datos debe permitir analizarla de forma pausada, y es deseable una representación gráfica.

## Consideraciones del entorno

No existen.

## Relaciones con otros sistemas

No existen.

## Restricciones generales

* El documento de diseño de proyecto se realizará siguiendo el formato establecido en el texto base y utilizando los diagramas de UML que se precisen: casos de uso, clases, secuencia, colaboración, estado y actividad. Se deberá utilizar alguna herramienta software para la representación UML, recomendándose ArgoUML.
* La codificación de la aplicación se realizará en lenguaje Java.

## Descripción del modelo.

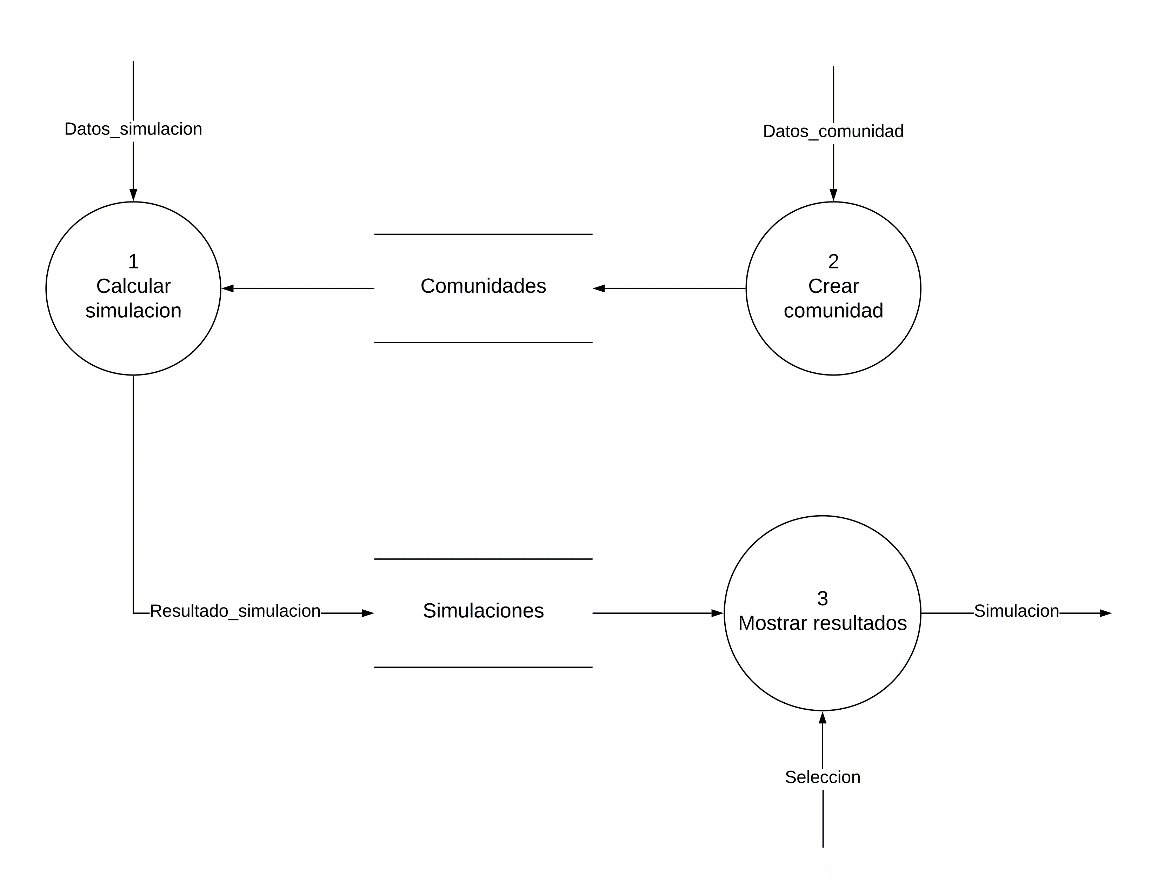
La descripción del modelo se va a realizar mediante la realización de diagramas de flujos de datos(DFD) y la descripción de los mismos con un diccionario de datos(DD). Dado el bajo nivel de complejidad en cuanto a la relación de los datos, se prescinde de una representación del modelo de datos más allá de la ofrecida por el DD.

### Diagramas de flujo de datos.

Imagen que contiene pájaro

Descripción generada automáticamenteLa organización principal del sistema, reflejada en el diagrama de contexto, implica únicamente a un usuario que introduce o modifica en un formulario los datos requeridos para la simulación y el sistema muestra el resultado por pantalla:

Diagrama de Contexto

La primera división interna distribuye las funciones en tres procesos principales, que se describen a continuación, aunque se especificarán en la siguiente sección *4.* *Requisitos Específicos*:

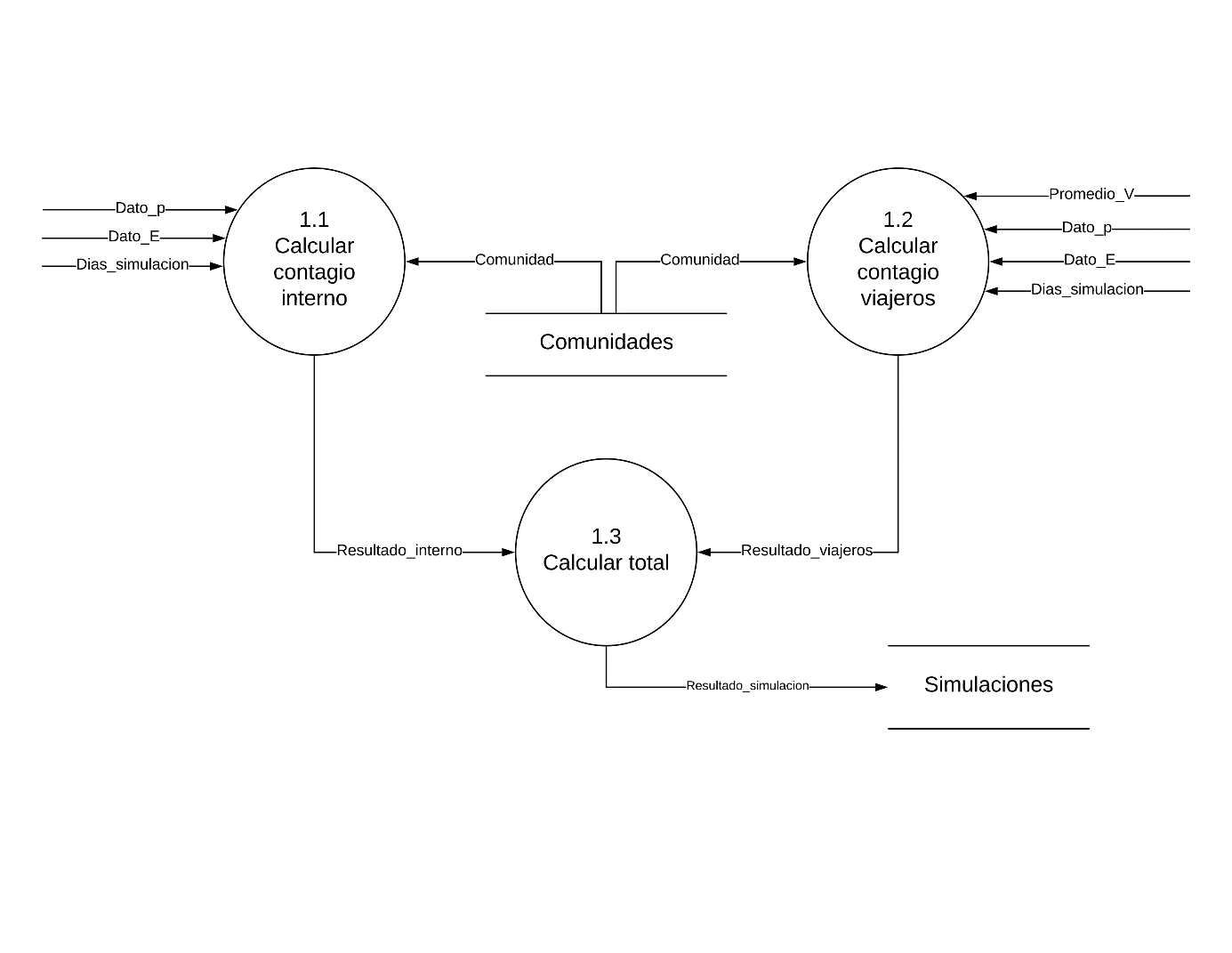
DFD 0. Simulador expansión de virus

*1*. Calcular simulación

Estas funciones realizan los cálculos para la simulación, que se desglosa en el DFD 1.

*Función 1.1* Calcular contagio interno: Aplica la fórmula de cálculo de contagio interno en una comunidad.

*Función 1.2* Calcular contagio viajeros: Aplica la fórmula de contagio por viajeros de otras comunidades.

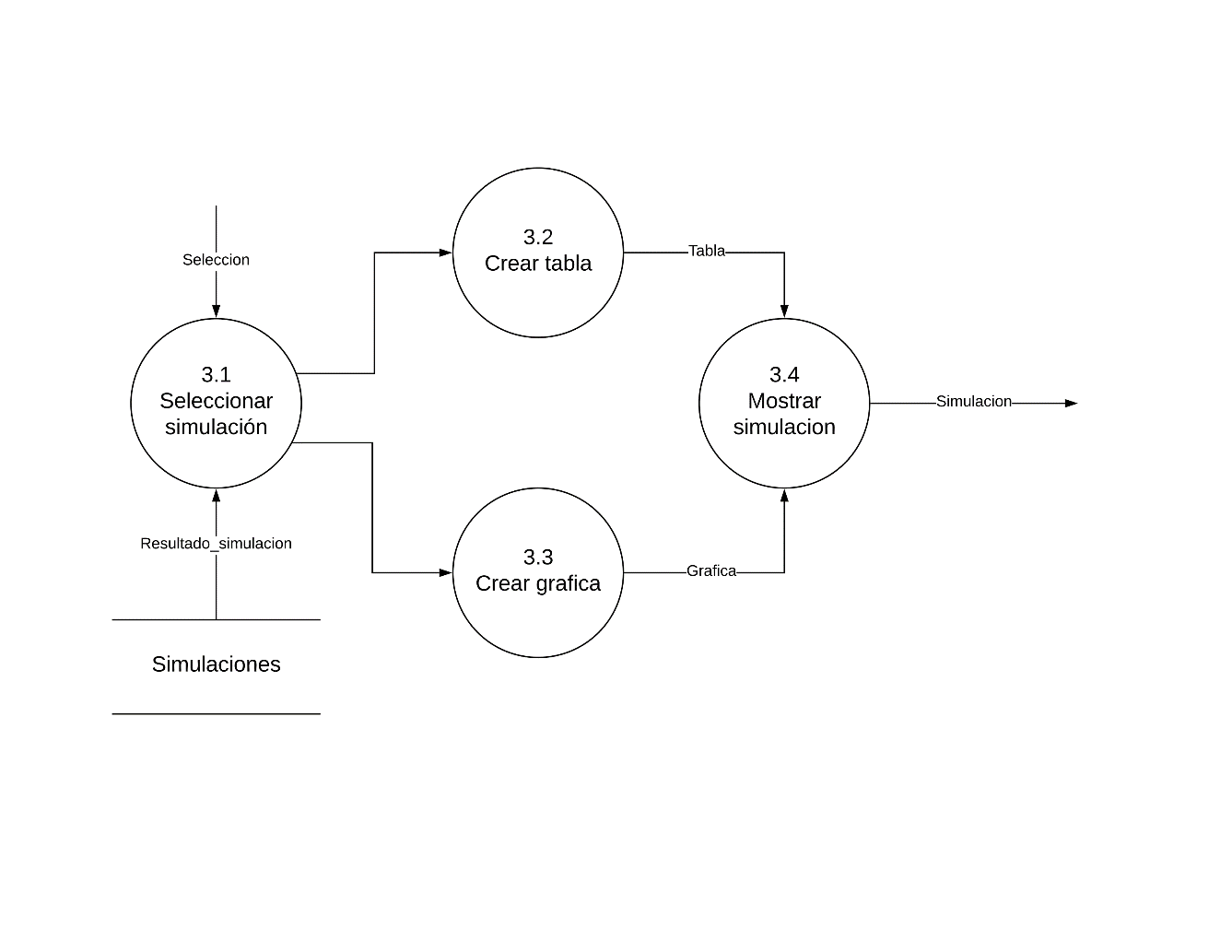
*Función 1.3* Calcular total: Suma y graba el resultado de la simulación.

DFD 1. Calcular simulación

*2.* Crear comunidad

Este proceso no se expande, ya que únicamente crea y graba las comunidades de acuerdo a los datos introducidos por el usuario.

*3.* Mostrar resultados

En este proceso se gestiona la representación de la simulación, permitiendo el requerido análisis pausado de los datos.

DFD 3. Mostrar resultados

*Función 3.1* Seleccionar simulación: Gestiona la selección de la simulación para mostrar.

*Función 3.2* Crear tabla: Configura la tabla para la representación.

*Función 3.3* Crear gráfica: Configura la gráfica para la representación.

*Función 3.4* Mostrar simulación: Configura la muestra por pantalla de la representación de las simulaciones.

### Diccionario de Datos.

A continuación se describen los elementos de datos utilizados en la descripción del sistema.

|  |  |  |
| --- | --- | --- |
|  | *Dato* | *Descripción* |
| Almacenes de Datos | COMUNIDADES | {COMUNIDAD} |
| SIMULACIONES | {ID\_SIMULACION + RESULTADO\_SIMULACION} |
| COMUNIDAD | ID\_COMUNIDAD + NOMBRE + TIPO + POBLACION + INFECTADOS |
| ID\_COMUNIDAD | [ 0 | .. | 9 ] + [ 0 | .. | 9 ]/Número entero de dos cifras/ |
| NOMBRE | /Cadena de caracteres/ |
| TIPO | [ PAIS | PROVINCIA | PUEBLO ] |
| POBLACION | /Número entero/ |
| INFECTADOS | /Número entero ≤ POBLACION/ |
| ID\_SIMULACION | /Número que referencia las distintas simulaciones realizadas en una sesión/ |
| RESULTADO\_SIMULACION | {ID\_COMUNIDAD + {DIA + {POBLACION + INFECTADOS}}/NUMERO\_COMUNIDADES/}/DIAS\_SIMULACION/ |
| DIA | /Número ordenado del día de la simulación/ |
| D.C. | DATOS | [DATOS\_SIMULACION + DATOS\_COMUNIDAD | SELECCIÓN] |
| SIMULACIÓN | TABLA + {GRAFICA} |
| DFD 0 | DATOS\_SIMULACION | DATO\_p + DATO\_E + PROMEDIO\_V + DIAS\_SIMULACION |
| DATOS\_COMUNIDAD | NUMERO\_COMUNIDADES + {POBLACION} |
| SELECCION | /Orden de representar una de las simulaciones realizadas en la sesión/ |
| NUMERO\_COMUNIDADES | /Número entero que especifica el total de comunidades de la simulación/ |
| DFD 1 | DATO\_p | /Número racional que representa la probabilidad de infectarse con un contacto/ |
| DATO\_E | /Número entero promedio de los contactos de un infectado con no infectados/ |
| PROMEDIO\_V | /Número racional porcentaje de habitantes de cada comunidad que viajan a otras/ |
| DIAS\_SIMULACION | /Número entero de los días a simular/ |
| RESULTADO\_INTERNO | {ID\_COMUNIDAD + {DIA + INFECTADOS\_INTERNOS}} |
| RESULTADO\_VIAJEROS | {ID\_COMUNIDAD + {DIA + INFECTADOS\_INTERNO}} |
| INFECTADOS\_INTERNOS | /Número entero total de infectados en un día por contagio interno/ |
| INFECTADOS\_VIAJEROS | /Número entero total de infectados en un día por contagio de viajeros/ |
| DFD 3 | TABLA | RESULTADO\_SIMULACION /Representación relacional/ |
| GRÁFICA | RESULTADO\_SIMULACION /Representación gráfica/ |

# Requisitos Específicos

A continuación se describen los requisitos específicos del sistema, que por defecto se entienden como obligatorios.

## Requisitos funcionales

*Función 1.1* Calcular contagio interno: Aplica la fórmula de cálculo de contagio interno en una comunidad.

* + Entrada: DATO\_p, DATO\_E, DIAS\_SIMULACION
  + Salida: RESULTADO\_INTERNO
  + Usa: COMUNIDADES
  + Actualiza:
  + Efecto: Crea una mapa de cada comunidad con los días de simulación y a su vez con los datos de población e infectados que resultan de la simulación.
  + Excepciones:

*Función 1.2* Calcular contagio viajeros: Aplica la fórmula de contagio por viajeros de otras comunidades.

* + Entrada: DATO\_p, DATO\_E, DIAS\_SIMULACION, PROMEDIO\_V
  + Salida: RESULTADO\_VIAJEROS
  + Usa: COMUNIDADES
  + Actualiza:
  + Efecto: Crea una mapa de cada comunidad con los días de simulación y a su vez con los datos de población e infectados que resultan de la simulación.
  + Excepciones:

*Función 1.3* Calcular total: Suma y graba el resultado de la simulación.

* + Entrada: RESULTADO\_INTERNO, RESULTADO\_VIAJEROS
  + Salida: RESULTADO\_SIMULACION
  + Usa:
  + Actualiza: SIMULACIONES
  + Efecto: Suma de forma coherente los resultados de las funciones 1.1 y 1.2 y graba la simulación.
  + Excepciones: Cuando INFECTADOS = POBLACIÓN se deja de incrementar la cifra.

*Función 2.0* Crear comunidad: Suma y graba el resultado de la simulación.

* + Entrada: DATOS\_COMUNIDAD
  + Salida: COMUNIDAD
  + Usa:
  + Actualiza: COMUNIDADES
  + Efecto: Crea un conjunto de comunidades según los datos introducidos por el usuario.
  + Excepciones:

*Función 3.1* Seleccionar simulación: Gestiona la selección de la simulación para mostrar.

* + Entrada: SELECCION
  + Salida: RESULTADO\_SIMULACION
  + Usa: SIMULACIONES
  + Actualiza:
  + Efecto: Recupera los resultados de la simulación seleccionada para su representación por pantalla.
  + Excepciones:

*Función 3.2* Crear tabla: Configura la tabla para la representación.

* + Entrada: RESULTADO\_SIMULACION
  + Salida: TABLA
  + Usa:
  + Actualiza:
  + Efecto: Crea una tabla ordenada con los resultados de la simulación por comunidades y días.
  + Excepciones:

*Función 3.3* Crear gráfica: Configura la gráfica para la representación. **(recomendable)**

* + Entrada: RESULTADO\_SIMULACION
  + Salida: GRAFICA
  + Usa:
  + Actualiza:
  + Efecto: Crea gráficas por comunidades de la evolución diaria de infectados.
  + Excepciones:

*Función 3.4* Mostrar simulación: Configura la muestra por pantalla de la representación de las simulaciones.

* + Entrada: GRAFICA, TABLA
  + Salida: SIMULACION
  + Usa:
  + Actualiza:
  + Efecto: Muestra por pantalla los resultados.
  + Excepciones:

# Apéndice único: Enunciado de la práctica

**Introducción**

Se presenta el enunciado de las prácticas de la asignatura Introducción a la Ingeniería del Software del Grado de Ingeniería Informática y Grado de Tecnologías de la Información. También se presentará las condiciones de entrega de los trabajos enunciados a los tutores, las condiciones de corrección y posterior entrega de calificaciones.

El contenido de estas prácticas integra dos partes progresivas que el alumno irá realizando y entregando a su tutor para su corrección.

**Objetivos**

El objetivo de esta práctica es la consolidación de los conocimientos adquiridos con el estudio de la asignatura mediante un ejercicio que abarca los contenidos del curso. Es importante haber estudiado con anterioridad los diferentes temas que integran el curso y haber realizado alguno de los ejercicios propuestos. La práctica está estructurada de tal forma que se puede ir realizando a medida que se van estudiando los temas y permite al alumno comprobar hasta qué punto ha asimilado los conocimientos. Un objetivo secundario de esta práctica es comparar como se aborda la realización de un proyecto software desde la óptica de la ingeniería frente al ya conocido y realizado desde la óptica del desarrollador o el programador.

**Enunciado**

Se desea realizar la formalización del proceso de desarrollo de una aplicación a partir de su especificación. Como especificación utilizaremos la descripción pormenorizada de la aplicación del apéndice A. Esta aplicación es un simulador de la expansión de un virus siguiendo un modelo exponencial.

Adicionalmente habrá que programar, orientado a objetos, la aplicación. El programa deberá ser totalmente consistente con el diseño realizado, es decir, no vale diseñar algo y programar otra cosa. Particularmente se pide

* Práctica 1: Elaborar el documento de diseño del proyecto especificado en el apéndice A. El documento debe seguir el formato establecido en el libro de texto recomendado para la asignatura, si bien debe utilizar los diagramas de UML que se precisen: casos de uso, clases, secuencia, colaboración, estado y actividad. Para la representación de los diagramas UML que se piden se deberá utilizar alguna herramienta software. Se recomienda utilizar Argo, que es de libre distribución, y puede descargarse en: http://argouml.tigris.org/
* Práctica 2: Programar en Java o cualquier otro lenguaje, previo acuerdo con el tutor, la aplicación de simulación de expansión de un virus siguiendo los documentos de diseño que se han entregado en la práctica 1. Van a usar sus propios “planos”.

Las decisiones que se han tomado en el diseño, así como las consideraciones que haya hecho el tutor respecto del diseño deberán verse reflejadas, EXPLICITAMENTE, en una pequeña memoria justificativa que acompañara la solución codificada. Se trata de ayudar a que el tutor pueda evaluar fácilmente el trabajo desarrollado y que compruebe que sus recomendaciones se han tenido en cuenta. Les ha tocado un cliente exigente que supervisa no sólo el producto sino la ingeniería del producto.

**Condiciones de presentación**

El calendario de presentación de las prácticas será el que marque el tutor, teniendo en cuenta que han de entregarse por orden. La práctica 2 estará sujeta a los comentarios y correcciones que el tutor indique de la práctica 1. Las fechas de entrega que aparecen en el enlace de tareas son una restricción mínima que el tutor del centro asociado puede restringir para favorecer la corrección de éstas.

Las prácticas se entregarán a través de la plataforma de UNED en “tareas”.

La entrega de una sola de las prácticas ya sea la 1 o la 2, no tendrá ningún valor de cara a la nota final del curso. Se deberán entregar las dos.

**APÉNDICE A**

Realizar un programa orientado a objetos en Java que simule la expansión de un virus en una población siguiendo un modelo exponencial.

* El número de casos de infectados depende de:
* El número de contactos que en promedio tenga cada infectado con personas no infectadas, lo llamaremos E.
* La probabilidad de infectarse con un contacto, que llamaremos p.

Si el número de infectados de una comunidad en un día d es Nd, el número de infectados el día siguiente Nd+1 será:

Nd+1 = Nd + Nd \*E\*p = Nd \* (1+E\*p)

Además, debemos pensar que vivimos en comunidades (ya sean países, pueblos o provincias) con una limitada interrelación unas con otras, lo que limita la expansión del virus de unas a otras.

Supongamos que V es el número de viajeros diarios de una comunidad a otra, entonces el número de infectados en una comunidad por culpa de los viajeros Nv sería:

Nv=E\*p\*V\*Nd,origen/Porigen

Donde Nd,origen sería número de infectados en la comunidad de origen del viajero y Porigen sería la población de la comunidad de origen. El cociente entre estos números representa la probabilidad de que el viajero esté infectado.

Por lo tanto habrá que sumar Nd+1 y Nv para saber el total de infectados en día d+1 en una comunidad.

El programa debe permitir introducir los siguientes datos a usuario del programa:

* Número de comunidades y la población de cada una de ellas.
* Porcentaje de habitantes de cada comunidad que viaja diariamente a cada una de las otras comunidades, igual para todas las comunidades para simiplificar.
* Coeficientes E, p y V.
* Número de días a simular

El programa debe devolver una tabla con los datos de infectados día a día en cada comunidad, el porcentaje de población infectada de cada comunidad, el número de infectados de la población total y el porcentaje de población total infectada. La presentación de los datos debe ser tal que se puedan analizar de forma pausada, apreciándose las representaciones gráficas.

Una vez terminada una simulación, el programa debe permitir modificar cualquiera de los parámetros y volver a realizar la simulación.

Se supone que el día uno aparece un primer infectado en una de las poblaciones.