

---

# **Documento de Especificación de Requisitos de Software (SRD)**

**para**

## **Simulador de expansión de virus**

**Versión 1.0**

**Sergio Flor**

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

**6 de abril de 2020**

## Tabla de Contenidos

<b>1. Introducción .....</b>	<b>1</b>
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
<b>2. Descripción General.....</b>	<b>2</b>
2.3 Objetivos y funciones .....	2
2.6 Restricciones generales.....	3
2.7 Descripción del modelo.....	4
2.7.1 Diagramas de flujo de datos. ....	4
2.7.2 Diccionario de Datos.....	6
2.7.3 Casos de Uso.....	7
<b>3. Requisitos Específicos .....</b>	<b>7</b>
3.1 Requisitos funcionales.....	8
3.4 Requisitos de operación.....	9
<b>Apéndice único: Enunciado de la práctica.....</b>	<b>11</b>

## Historial de Revisiones

Nombre	Fecha	Razón de los cambios	Versión

# 1. Introducción

## 1.1 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.

## 1.2 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.

## 1.3 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:
  - **N<sub>d</sub>** : número de infectados un día concreto.
  - **N<sub>d+1</sub>** : Número de infectados al día siguiente.
  - **N<sub>v</sub>** : Número de infectados por culpa de los viajeros.

## 1.4 Referencias

- R1. Enunciado de las prácticas curso 2019-2020 para la asignatura Introducción a la Ingeniería de Software.
- R2. Gómez Palomo, S., & Moraleda Gil, E. (2020). *Aproximación a la Ingeniería de Software* (segunda ed.). Madrid: Ramon Areces.

## 1.5 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.

# 2. Descripción General

## 2.1 Relación con otros proyectos Ninguna

## 2.2 Relación con proyectos anteriores y posteriores Ninguna

## 2.3 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 de forma interna en una comunidad según la fórmula:

$$N_{d+1} = N_d + N_d * E * p = N_d * (1 + E * p)$$

- El número de infectados en una comunidad por culpa de los viajeros con la fórmula:

$$N_v = E * p * V * (N_{d,origen} / P_{origen})$$

Donde  $N_{d,origen}$  sería número de infectados en la comunidad de origen del viajero y  $P_{origen}$  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 se describe como el número de viajeros y por tanto se calcula en base al porcentaje del punto anterior sobre la población)
- Número de días a simular.

A partir de estas características, donde se especifica que los “viajeros” de una comunidad viajan **todos los días y a cada una de las otras comunidades**, vemos que habría **que desarrollar la fórmula del número de infectados por viajeros** de otras comunidades, ya que tal como aparece en el enunciado, es aplicable sólo para una simulación con dos comunidades. Por tanto, se realizaría el cálculo de la siguiente manera:

$$N_v = E * p * \text{SUMA} \{ \text{PARA CADA (Comunidad externa)} \text{ HACER } (V * (N_{d,\text{origen}} / P_{\text{origen}})) \}$$

Estos datos:

- Deben poder introducirse por primera vez.
- 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. Siendo una tabla, la presentación de los datos es estática en pantalla y permite usar el tiempo que se quiera para observarla. Si embargo, el hecho de especificar “de forma pausada” puede tener dos implicaciones:
  - Se requiere que la salida de los datos pueda realizarse con pausas.
  - El hecho de poder “modificar” los datos de la simulación tiene como objeto poder alternar la presentación de los distintos resultados para analizarlos.
- Es deseable una representación gráfica. Se supone pues, en forma de algún tipo de gráficos para analizar la evolución de los resultados que se piden: infectados y porcentajes de población infectada

## 2.4 Consideraciones del entorno No existen.

## 2.5 Relaciones con otros sistemas No existen.

## 2.6 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.

## 2.7 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.

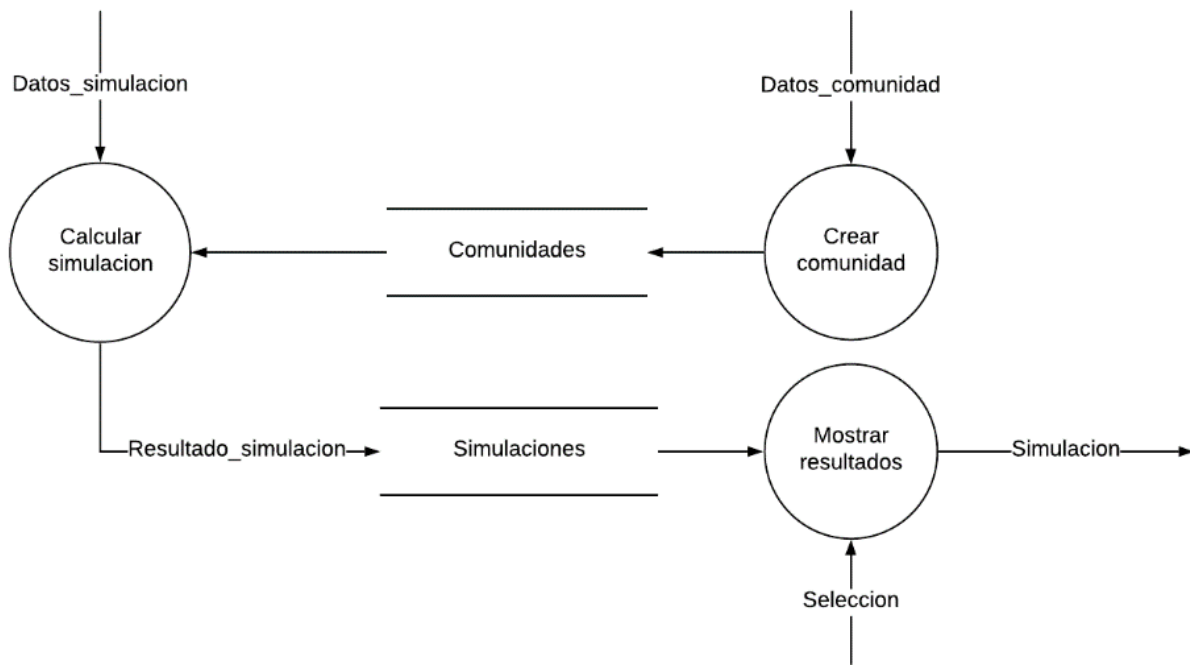
### 2.7.1 Diagramas de flujo de datos.

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



Diagrama de Contexto

El primer desarrollo del sistema distribuye las funciones en tres procesos principales, que se describen a continuación, aunque se especificarán en la siguiente sección 4. *Requisitos Específicos*:

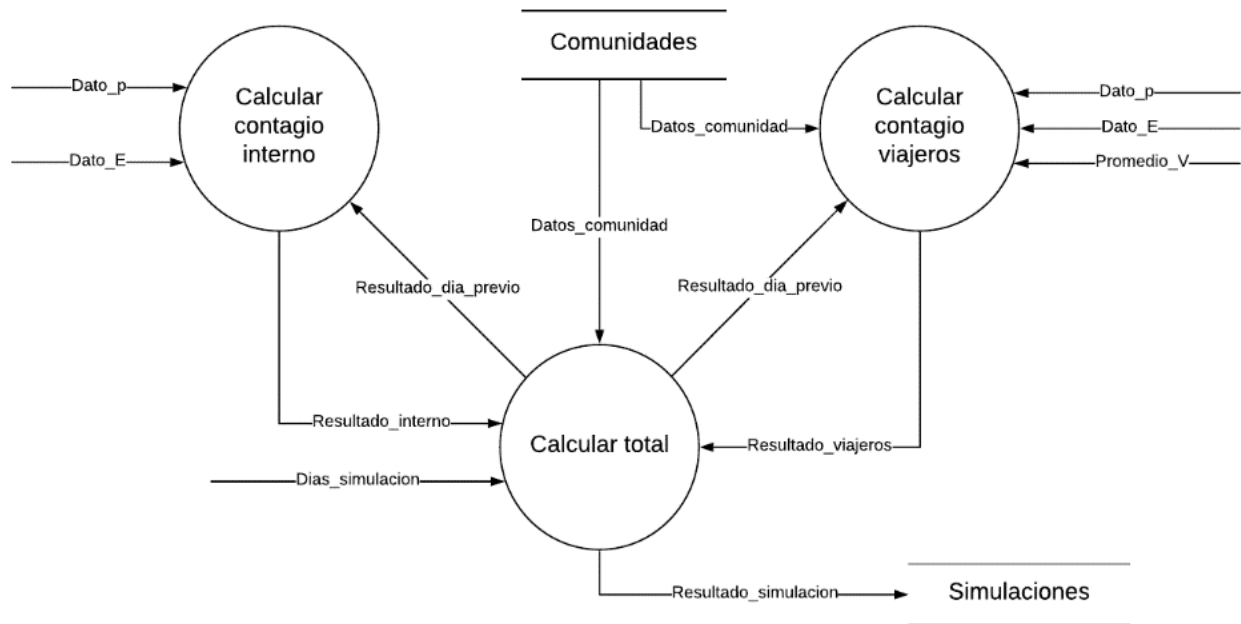


DFD 0. Simulador expansión de virus

- Calcular simulación

Estas funciones realizan los cálculos para la simulación, que se desglosa en el DFD 1. Se realizan sobre los datos que ha introducido o modificado el usuario.

- Calcular contagio interno: Aplica la fórmula de cálculo de contagio interno en una comunidad.
- Calcular contagio viajeros: Aplica la fórmula de contagio por viajeros de otras comunidades.
- Calcular total: Suma y graba el resultado de la simulación para cada día de simulación.

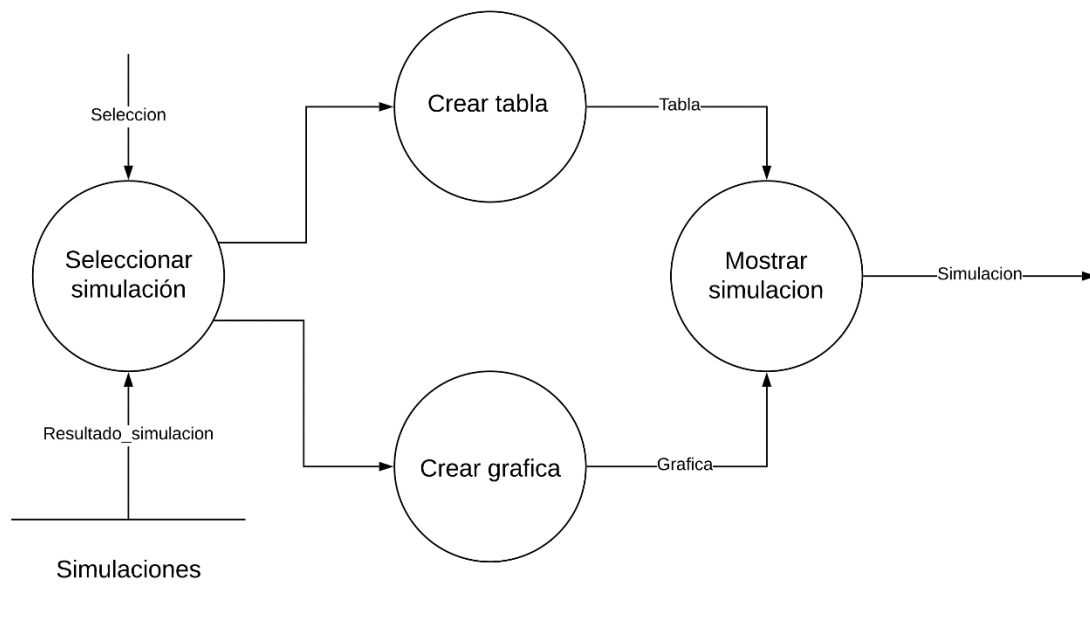


DFD 1. Calcular simulación

- Crear comunidad

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

- Mostrar resultados



DFD 3. Mostrar resultados

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

- Seleccionar simulación: Gestiona la selección de la simulación para mostrar.
- Crear tabla: Configura la tabla para la representación.
- Crear gráfica: Configura la gráfica para la representación.
- Mostrar simulación: Configura la muestra por pantalla de la representación de las simulaciones.

### 2.7.2 Diccionario de Datos.

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

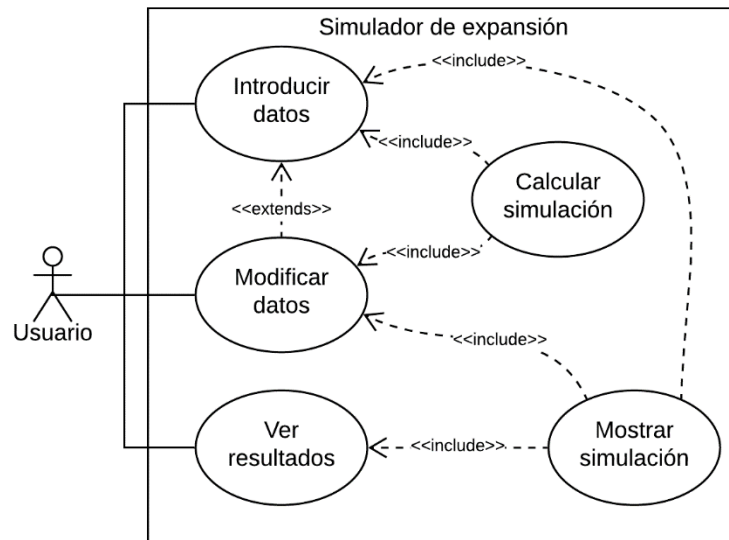
	DATO	DESCRIPCIÓN
Almacenes de Datos	COMUNIDADES	{COMUNIDAD}
	SIMULACIONES	{RESULTADO_SIMULACION}
	COMUNIDAD	NOMBRE + TIPO + POBLACION
	NOMBRE	/Cadena de caracteres/
	TIPO	[ COMUNIDAD   PAIS   PROVINCIA   PUEBLO ]
	POBLACION	/Número entero/
	RESULTADO_SIMULACION	{ COMUNIDAD + {FECHA + INFECTADOS} / DIAS_SIMULACION / } / NUMERO_COMUNIDADES /
	FECHA	/Fecha de un día de la simulación/
D.C.	DATOS	[DATOS_SIMULACION + DATOS_COMUNIDAD   SELECCIÓN]
	SIMULACIÓN	TABLA + {GRAFICA}
DFD 0	DATOS_SIMULACION	DATO_p + DATO_E + DATO_V + DIAS_SIMULACION
	DATOS_COMUNIDAD	{NOMBRE + POBLACION} / NUMERO_COMUNIDADES /
	SELECCION	/Orden de representar una de las simulaciones realizadas en la sesión/
	NUMERO_COMUNIDADES	/Número 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/
	DATO_V	/Número racional porcentaje de habitantes de cada comunidad que viajan a otras/
	DIAS_SIMULACION	/Número entero de los días a simular/
	RESULTADO_DIA_PREVIO	/Número entero total de infectados en el día previo/
	RESULTADO_INTERNO	{COMUNIDAD + {FECHA + INFECTADOS_INTERNOS}}
	RESULTADO_VIAJEROS	{ COMUNIDAD + {FECHA + INFECTADOS_VIAJEROS}}
	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 de los resultados/
	GRÁFICA	RESULTADO_SIMULACION /Representación en gráficas de infectados y porcentajes/



### 2.7.3 Casos de Uso

La descripción de los casos de uso en un programa tan específico parece obvia, pero permite centrar la atención en los dos tipos interacciones que tendrá el usuario, que será el único actor, con el programa:

- La introducción/modificación de datos para el cálculo.
- El análisis de los resultados mostrados por pantalla.



#### 2.7.3.1 Caso de Uso: Introducir datos

- El usuario introduce los datos de las comunidades y los parámetros para generar una nueva simulación.
- El programa crea el almacén de comunidades y genera cada comunidad según los datos introducidos.
- El programa configura la simulación según los parámetros y la calcula sobre los datos de las comunidades.
- El programa almacena los resultados de la simulación y los muestra por pantalla.

#### 2.7.3.2 Caso de Uso: Modificar datos («extends» Introducir datos)

- Sobre los datos de la simulación anterior, el usuario modifica los parámetros que considere y se vuelve a generar una nueva simulación, siguiendo la misma secuencia del caso de uso extendido.

#### 2.7.3.3 Caso de Uso: Ver resultados

- El usuario analiza las diferencias provocadas por las modificaciones que ha ido realizando sobre la simulación inicial (p. ej. mediante un sistema de anterior/siguiente), seleccionando los resultados de las sucesivas simulaciones.
- El programa muestra por pantalla los resultados de la simulación seleccionada.

## 3. Requisitos Específicos

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

## 3.1 Requisitos funcionales

3.1.1 El programa debe realizar la simulación sobre unas comunidades introducidas por usuario, para lo que contemplará la función:

*3.1.1.1 Crear comunidad: Crea y graba las comunidades y sus poblaciones, introducidas por el usuario.*

- Entrada: DATOS\_COMUNIDAD
- Salida:
- Usa:
- Actualiza: COMUNIDADES
- Efecto: Crea y graba un conjunto de comunidades.
- Excepciones:

3.1.2 El enunciado menciona la especialización de comunidades en países, provincias o pueblos (como inciso aclaratorio entre paréntesis) aunque describe todas las operaciones de la simulación utilizando comunidades “genéricas”. Por tanto, parece recomendable dotar al programa de una estructura básica y funciones que atiendan esta especialización para facilitar el desarrollo de este requerimiento cuando sea necesario. **(recomendable)**

3.1.3 El programa debe realizar las siguientes funciones para los cálculos de la simulación:

*3.1.3.1 Calcular contagio interno: Aplica la fórmula de cálculo de contagio interno en una comunidad para un día.*

- Entrada: DATO\_p, DATO\_E
- Salida: RESULTADO\_INTERNO
- Usa: RESULTADO\_DIA\_PREVIO
- Actualiza:
- Efecto: Provee al cálculo total del número de infectados por contagio interno.
- Excepciones:

*3.1.3.2 Calcular contagio viajeros: Aplica la fórmula de contagio por viajeros de otras comunidades para un día.*

- Entrada: DATO\_p, DATO\_E, DATO\_V
- Salida: RESULTADO\_VIAJEROS
- Usa: RESULTADO\_DIA\_PREVIO
- Actualiza:
- Efecto: Provee al cálculo total del número de infectados por contagio de viajeros de otras comunidades.
- Excepciones:

*3.1.3.3 Calcular total: Suma los resultados diarios provenientes de las funciones parciales anteriores para todos los días de la simulación, establecidos por la entrada, y controla el rebose de infectados respecto a la población.*

- Entrada: RESULTADO\_INTERNO, RESULTADO\_VIAJEROS
- Salida: RESULTADO\_SIMULACION
- Usa: DATOS\_COMUNIDAD
- 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.

3.1.4 El programa debe devolver una tabla con los resultados de infectados y los porcentajes respecto a la población de cada comunidad y del total de todas, día a día. Para lo que realizará las funciones:

*3.1.4.1 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:

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

- Entrada: TABLA, (GRAFICA)
- Salida: SIMULACION
- Usa:
- Actualiza:
- Efecto: Muestra por pantalla los resultados.
- Excepciones:

**3.2 Requisitos de capacidad** No se especifican.

**3.3 Requisitos de interfase** No se especifican.

### **3.4 Requisitos de operación**

3.4.1 El programa debe permitir al usuario introducir los datos requeridos y realizar modificaciones sobre los parámetros, por lo que debe ofrecerse una interfaz adecuada para poder realizar estas operaciones facilitando la modificación de un o unos parámetros y la conservación de los valores previos en otros.

3.4.2 La asignación del porcentaje de habitantes de cada comunidad que viaja a otras comunidades se introduce igual para simplificar, lo que implica que sería deseable en una versión más compleja poder trabajar con distintos porcentajes para cada comunidad, y por tanto permitir su introducción. **(opcional)**

3.4.3 El programa debe presentar los datos de forma que se puedan analizar de forma pausada, para ello puede ser deseable permitir recuperar los resultados de las distintas simulaciones realizadas. **(opcional)**

3.4.4 Se aprecia también presentar el resultado de forma gráfica para lo que el programa debería de realizar gráficos que representan la evolución diaria de los distintos datos que se piden, pudiéndose mostrar por ejemplo, en dos gráficas que agrupen las comunidades y el total de la evolución de infectados por un lado y de la evolución de porcentajes por otro. **(recomendable)**

**3.5 Requisitos de recursos** No se especifican.

**3.6 Requisitos de verificación** No se especifican.

**3.7 Requisitos de pruebas de aceptación** No se especifican.

**3.8 Requisitos de documentación** No se especifican.

**3.9 Requisitos de seguridad** No se especifican.

**3.10 Requisitos de transportabilidad** No se especifican.

**3.11 Requisitos de calidad** No se especifican.

**3.12 Requisitos de fiabilidad** No se especifican.

**3.13 Requisitos de mantenibilidad** No se especifican.

**3.14 Requisitos de salvaguarda** No se especifican.

## 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  $N_d$ , el número de infectados el día siguiente  $N_{d+1}$  será:

$$N_{d+1} = N_d + N_d * E * p = N_d * (1 + E * p)$$

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  $N_v$  sería:

$$N_v = E * p * V * N_{d, \text{origen}} / P_{\text{origen}}$$

Donde  $N_{d, \text{origen}}$  sería número de infectados en la comunidad de origen del viajero y  $P_{\text{origen}}$  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  $N_{d+1}$  y  $N_v$  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 simplificar.
- 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.