**UNIVERSIDAD DE LIMA**

**ESCUELA UNIVERSITARIA DE INGENIERÍA**

**CARRERA DE INGENIERÍA DE SISTEMAS**



***Tarea 1***

*Sección 703*

*INGENIERIA DE SOFTWARE I*

*Profesor: JUAN MANUEL GUTIERREZ CARDENAS*

**Integrante:**

**MANDUJANO NIMA RICARDO JESUS 20100661**

**Abril 2017**

1. Opinión lectura Dijkstra:

Edsger W. Dijkstra en su artículo invita a todas las personas a tener una idea y pensar una manera de ensenar la programación como un curso introductorio según sus palabras “*Desde el comienzo, y a través de todo el curso, enfatizamos que la tarea del programador no es sólo escribir un programa, sino que su tarea principal es dar una prueba formal…*”, pero a pesar de esto en el caso de la Ingeniería de Software nos dice que esta representa la forma de como atacar un problema complejo y llegar a la solución más adecuada la cual envuelve mucho más que tan solo escribir unas líneas de código de programación.

Para Dijkstra, como expone en su obra sobre las ciencias de la computación, es necesario abandonar una metáfora que afecta a las computadoras actuales y su misma programación. Ademas nos dice que la Ingenieria de Software se presenta como una causa meritoria descubriendo que el lema de “como programar si no puede”.

1. Casos de Crisis de Software.

**Apagón de EE.UU 2003:**

El 14 de agosto del 2003 se dio uno de los apagones más grandes en la historia afectando alrededor de 55 millones de personas en diversas ciudades de EE.UU (New York, New Jersey, entre otros) y Canadá (Ontario). El suceso empezó con la caída de una central eléctrica cerca de Lake Erie en Ohio lo que puso en alto estrés y comprometió a toda la red eléctrica, sucede que cuando las líneas se encontraban bajo carga eléctrica estas se calentaban más de lo normal afectando árboles y poniendo el sistema bajo mucho más presión, al darse un efecto en cascada de este evento la red eléctrica sufrió una reducción en 20% de su output normal.

Todo este suceso puedo haberse evitado o en el mejor de los casos reducir el efecto negativo en la población si no fuera por un error en el software del sistema general de alarma del centro de control, este escenario es llamado “Condición de carrera” donde hay 2 partes de un sistema que compiten por resolver el mismo problema sin ponerse de acuerdo, esto provoco un error en la alarma dejando de procesar alertas al personal de la planta, por desgracia la alarma no alerto a ningún encargado de que tenía un error.

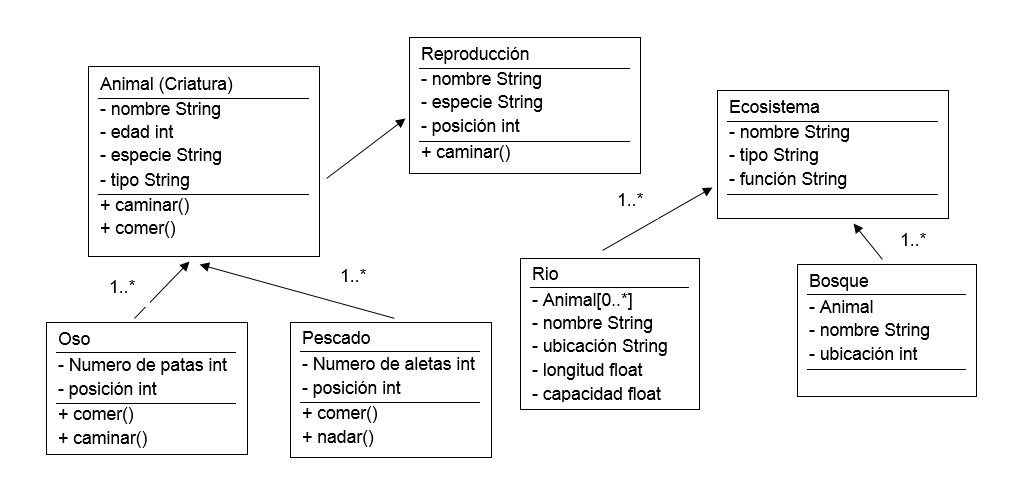
**El bug del milenio 2000:**

El problema del año 2000 o Y2K en los sistemas de computadoras se empleaba dos dígitos para mostrar la fecha por ejemplo 98 en vez de 1998, pero pocos anticipaban del problema que podría ocurrir llegado el año 2000 puesto que usando los sistemas de esa época el año 2000 solamente se podía representar como ‘00’ lo cual hacia que las computadoras se confundieran pensando que hacía referencia al año 1900, esto provocaría errores de cálculo en las fechas pasando el milenio.

Como respuesta las compañías de software tuvieron que adaptar y actualizar sus componentes que abarcaban desde los bancos hasta ventas de tickets de trenes, además se creó Y2K Cooperation Centre para coordinar y ayudar a las organizaciones en este tema y prepararlas para la llegada del nuevo milenio.

1. Diagramas UML:

**Caso 1:**



**Caso 2:**

