

# Ingeniería de Software

## Trabajo Práctico 1

### Documento de Requerimientos

Alumnos:  
Blanco, Lucas  
Murillo, Sebastian

Docente:  
Miceli, Martin

FCEFYN – UNC  
2019

## Índice:

Introducción -----	Pág. 3
Descripción General-----	Pág. 4
Requerimientos de Hardware y Software -----	Pág. 5
Lista de Requerimientos-----	Pág. 6
Requerimientos Detallados -----	Pág. 7
Diagrama de Caso de Usos-----	Pág. 9
Diagrama de Actividades -----	Pág. 10
Pruebas de Sistema -----	Pág. 11
Matriz de Trazabilidad -----	Pág. 12
Historial de Versiones -----	Pág. 13
Bibliografía -----	Pág. 14

## **1.- Introducción:**

Este documento presenta las funciones y requerimientos del Simulador de Poblaciones de Vida Salvaje. Así como también diagramas UML para graficar su uso, además de la matriz de trazabilidad entre requerimientos y casos de usos.

## **2.- Descripción General:**

El Simulador de Poblaciones de Vida Salvaje es un software interactivo que se utiliza para el control de especies en un espacio determinado a partir de la definición numérica de individuos de cada una.

El control se realiza entre dos especies entre las cuales cada una cumple un rol específico en la cadena alimentaria comportándose una como consumidor de la otra según el caso que se presente.

Con éste software un usuario podrá simular casos numéricos existentes, o probar distintos casos buscando un fin específico.

### **2.1.- Funcionalidad General:**

El Simulador de Poblaciones de Vida Salvaje tiene como función general ayudar a criadores de animales a simular casos existentes o buscar la cantidad específica de animales de cada especie posibles en un terreno determinado que pueden coexistir sin la extinción de una de ellas.

Ésto es posible de calcular a partir de reglas básicas de depredación naturales según las especies analizadas, y la cantidad de animales de esa especie en conjunto en cierto espacio comparada con la cantidad de la otra especie.

### **2.2.- Grupos de Usuarios:**

El software está destinado a usarse por dos grupos de usuarios.

El primer grupo, está compuesto por aquellos que configuren los parámetros básicos de funcionamiento del software como son las reglas de depredación entre especies, tiempos de vida de cada uno, características biológicas, entre otros.

El segundo grupo, está compuesto por aquellos usuarios finales que utilizaran el software para simular y realizar cálculos sobre especies definidas, pudiendo configurar tamaños de terrenos de interacción, cantidad de animales de cierta especie, tiempo de simulación, entre otros.

### **3.- Requerimientos de Software y Hardware:**

- Para poder utilizar el Simulador, es necesaria una computadora personal de escritorio, o una notebook, con hardware actualizado para la posibilidad de correr gráficos a alta velocidad.
- Para poder utilizar el Simulador, es necesario una computadora personal de escritorio, o notebook con sistema operativo Windows o Linux.

#### 4.- Lista de Requerimientos:

El software cumple con la siguiente lista de requerimientos funcionales:

**REQF. 1:** Ingreso de parametros iniciales – El sistema debe permitir el ingreso de datos característicos de cada especie.

**REQF. 2:** Ingreso de cantidad de animales – El sistema debe permitir el ingreso de la cantidad de animales que se van a simular.

**REQF. 3:** Pantalla de terreno y especies – El sistema debe mostrar gráficamente el terreno y la cantidad de animales que se esta simulando.

**REQF. 4:** Tiempo de simulación – El sistema debe permitir configurar, controlar y simular distintos tiempos de evolución.

**REQF. 5:** Aviso de extinción – El sistema debe avisar cuando una especie se extingue en una simulación.

**REQF. 6:** Aviso de estado final – El sistema debe avisar cuando la simulación llega a su tiempo final y cuales son las cantidades de cada especie.

El software cumple con la siguiente lista de requerimientos no funcionales:

**REQN. 1:** Aceptabilidad y accesibilidad – El sistema debe ser práctico y accesible al uso para cualquier cliente.

**REQN. 2:** Rendimiento – El sistema debe responder en tiempo y no colgarse.

**REQN. 3:** Confiabilidad – El sistema debe dar datos precisos y no generar errores en los cálculos que realice.

## **5.- Requerimientos detallados:**

### **5.1.- REQF. 1: Ingreso de Parámetros Iniciales:**

#### **- Descripción:**

El software debe poseer una interfaz para aquellos usuarios que deseen observar características biológicas de una especie, o quieran agregar una nueva.

El usuario debe poder ingresar datos biológicos como nombre de la especie, tiempo de vida, reproducción, muerte, entre otras que caractericen a ésta.

El software debe mostrar todos éstos datos, y aquellos relacionados a una especie.

### **5.2.- REQF. 2: Ingreso de Cantidad de Animales:**

#### **- Descripción:**

El software debe poseer una interfaz para aquellos usuarios finales que deseen realizar una simulación, donde puedan elegir las especies que interaccionaran en el terreno, y la cantidad de animales de cada especie.

El usuario debe poder ingresar datos numéricos de cada especie.

### **5.3.- REQF. 3: Pantalla de Terreno y Especies:**

#### **- Descripción:**

El software debe poseer una interfaz donde se pueda observar la interacción entre las especies en el terreno a medida que transcurre el tiempo, y la cantidad de animales que hay, sea por nacimiento o muerte, en un tiempo dado.

El software debe mostrar gráficamente ésta interacción marcando con unidades donde se ubica cada animal en cierto terreno delimitado.

### **5.4.- REQF. 4: Tiempo de Simulación:**

#### **- Descripción:**

El software debe poseer en la interfaz de interacción entre especies el tiempo transcurrido desde el inicio, y el tiempo en el que se finalizara la simulación, para que los usuarios puedan controlar éste.

El software debe mostrar un reloj.

### **5.5.- REQF. 5: Aviso de Extinción:**

#### **- Descripción:**

El software debe mostrar en la interfaz de interacción un aviso cuando una especie se extinga.

#### **5.6.- REQF. 6: Aviso Estado Final:**

- Descripción:

El software debe mostrar en una interfaz, cuál ha sido el estado de finalización de la simulación con los siguientes datos: tiempo simulado, cantidad de animales de la especie 1, cantidad de animales de la especie 2.

#### **5.7.- REQN. 1: Aceptabilidad y Accesibilidad:**

- Descripción:

Los usuarios que utilicen la aplicación no deben tener conocimientos específicos de como se realizan las simulaciones o como se comportan las distintas especies.

El software debe ser simple para que, los usuarios sin conocerlo, puedan ingresar los datos en los casilleros correspondientes, y puedan simular sin ningún inconveniente.

El escenario final, debe ser una fotografía exacta de como puede terminar un escenario real con las distintas cantidades de especies. Esto se logrará, teniendo en cuenta el tamaño del campo donde viven los animales, y la cantidad de cada especie.

#### **5.8.- REQN. 2: Rendimiento:**

- Descripción:

El sistema debe responder instantáneamente ante las solicitudes del usuario, tal como el ingreso de los valores, la selección de especies o el inicio de la simulación.

Así también el tiempo de simulación debe corresponder con el tiempo que se ha elegido para simular.

#### **5.9.- REQN. 3: Confiabilidad:**

- Descripción:

Los datos que devuelva y simule el software deben ser datos exactos para el propósito del mismo, por ejemplo: los comportamientos de las distintas especies deben corresponder con los datos característicos de ellas. Ya que, en base a los resultados los usuarios tomaran decisiones para sus aplicaciones.



## 6.- Diagramas:

Diagrama de Casos de Uso:

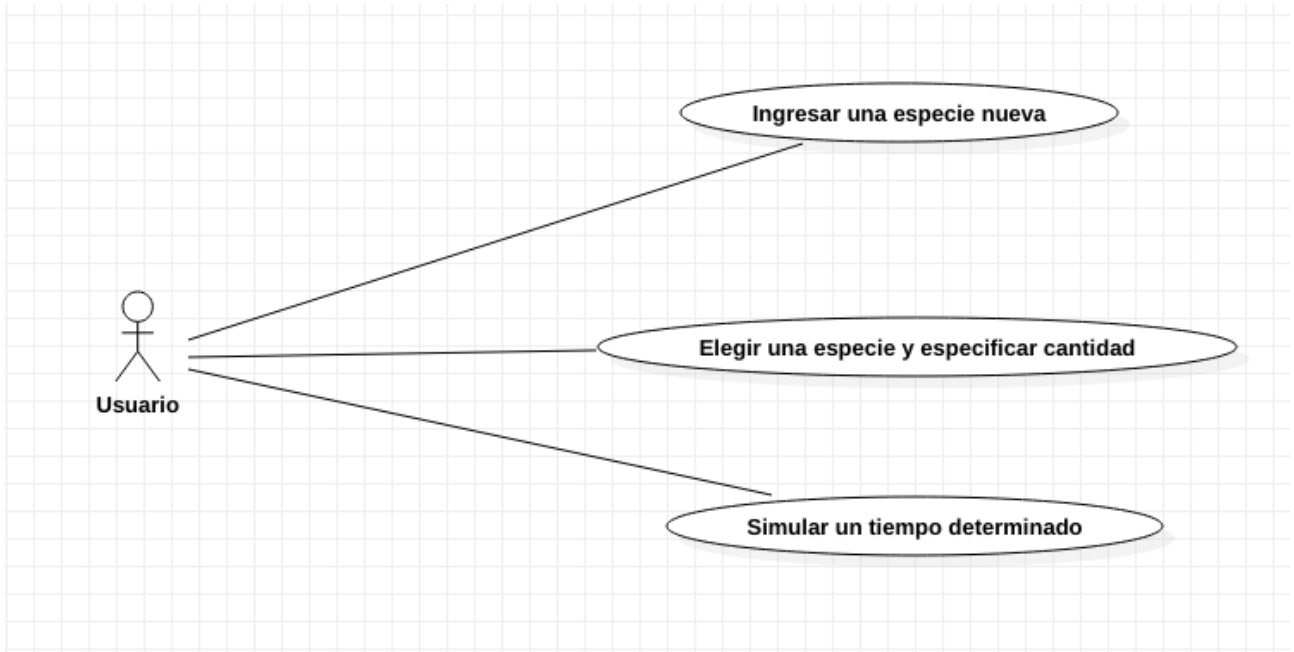
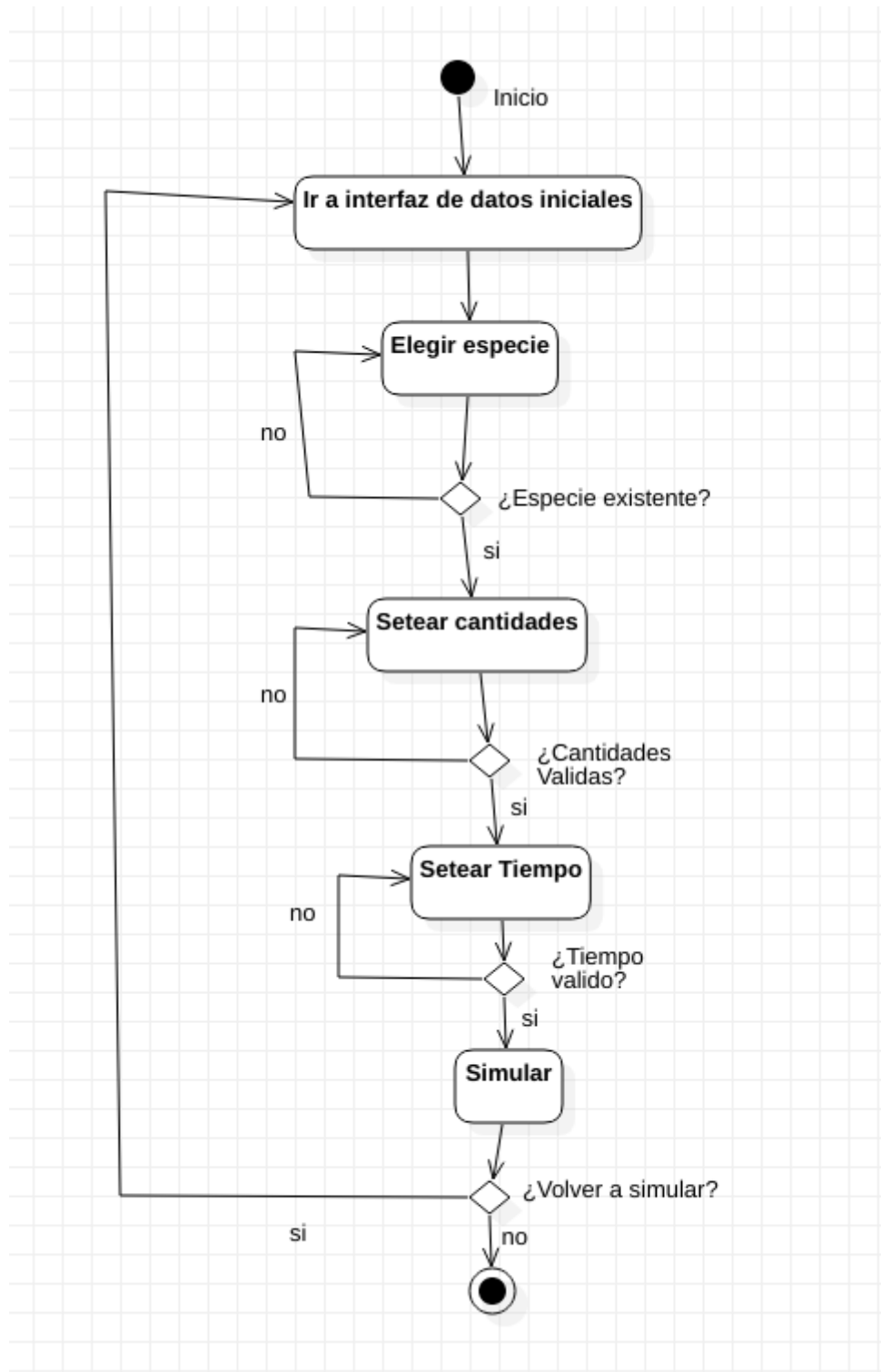


Diagrama de Actividades:

En el siguiente diagrama modelamos el caso particular en el que un usuario inicia la aplicación, elige especies de animales de la biblioteca de especies, ingresa sus cantidades, e inicia la simulación con el software.



## **7.- Pruebas de Sistema:**

### **Prueba de Sistema N°1:**

- Abrir aplicación
- Ingresar a interfaz de carga de datos
- Hacer click en el boton aceptar

Resultado Esperado: La aplicación nos dirige a la interfaz de carga de datos iniciales.

### **Prueba de Sistema N°2:**

- En la interfaz de selección
- Seleccionar especie y cantidades de animales
- Hacer click en el boton aceptar

Resultado Esperado: La aplicación nos carga la especie y cantidades.

### **Prueba de Sistema N°3:**

- En la interfaz de selección
- Setear tiempo de simulación
- Hacer click en aceptar

Resultado Esperado: La aplicación debe setear el tiempo que va a durar la simulacion.

### **Prueba de Sistema N°4:**

- Apretar boton de inicio de simulacion con datos cargados

Resultado Esperado: La aplicación corre el escenario con las cantidades cargadas y simula a medida que se disminuye el tiempo que va a tardar.

### **Prueba de Uso Normal N°1:**

- En interfaz de configuracion de datos
- Seleccionar cantidad de especie mayor al terreno existente
- Hacer click en el boton aceptar

Resultado Esperado: La aplicación nos da un mensaje que no puede cargar datos mayores al tamaño del terreno.

### **Prueba de Uso Normal N°2:**

- En interfaz de configuracion de datos
- Seleccionar cantidad de especie menor a cero.
- Hacer click en el boton aceptar

Resultado Esperado: La aplicación nos da un mensaje que no puede cargar datos negativos.

## **8.- Smoke Test:**

### **Smoke Test N°1:**

- Iniciar correctamente la aplicación, logrando ingresar en su pantalla inicial.

### **Smoke Test N°2:**

- Poder elegir especies, y cargar datos.

### **Smoke Test N°3:**

- Iniciar simulación de poblaciones.

### **Smoke Test N°4:**

- Obtener un resultado final de una simulación.

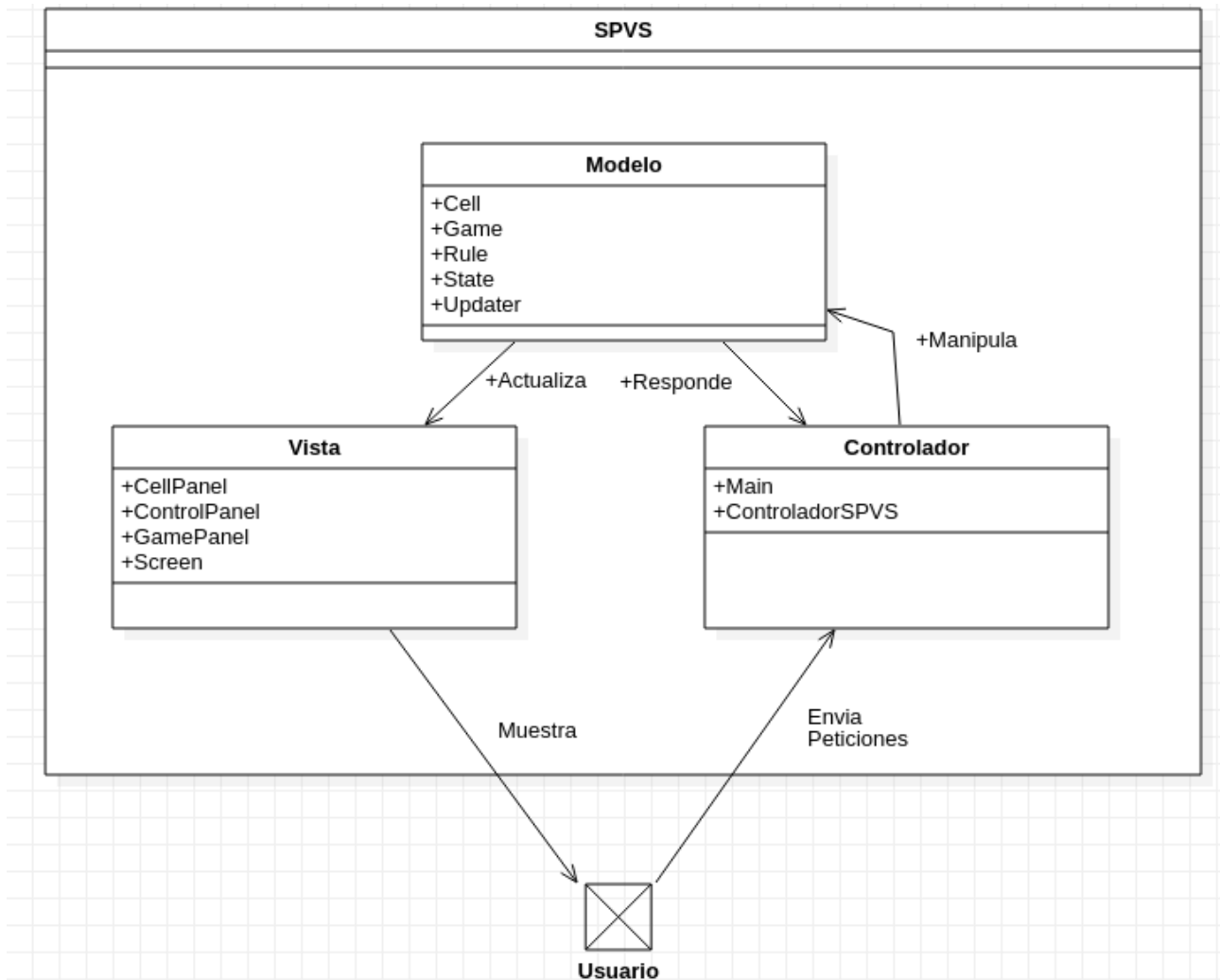
## 9.- Matriz de Trazabilidad:

Matriz entre los requerimientos del Software y si el requerimientos debe cumplirse por el software o debe esperarse algún dato del usuario.

	Simulador Poblacional	VistaSimulador	Caso de Uso 1	Caso de Uso 2	Caso de Uso 3
Req.1		x	x		
Req.2		x		x	
Req.3	x				x
Req.4	x				x
Req.5	x				x
Req.6	x				x

## 10.- Arquitectura del Sistema:

En el siguiente gráfico vemos la arquitectura de nuestro sistema, donde detallamos a que módulo pertenece cada clase que lo compone.



Al ser una aplicación que posee distintas tareas bien definidas, separaremos éstas en distintos módulos.

Así generaremos que la lógica del sistema esté separada de las vistas permitiendo que los datos cambien de manera independiente de su presentación y viceversa.

Por este motivo, seleccionamos el patrón MVC (Model-View-Controller).

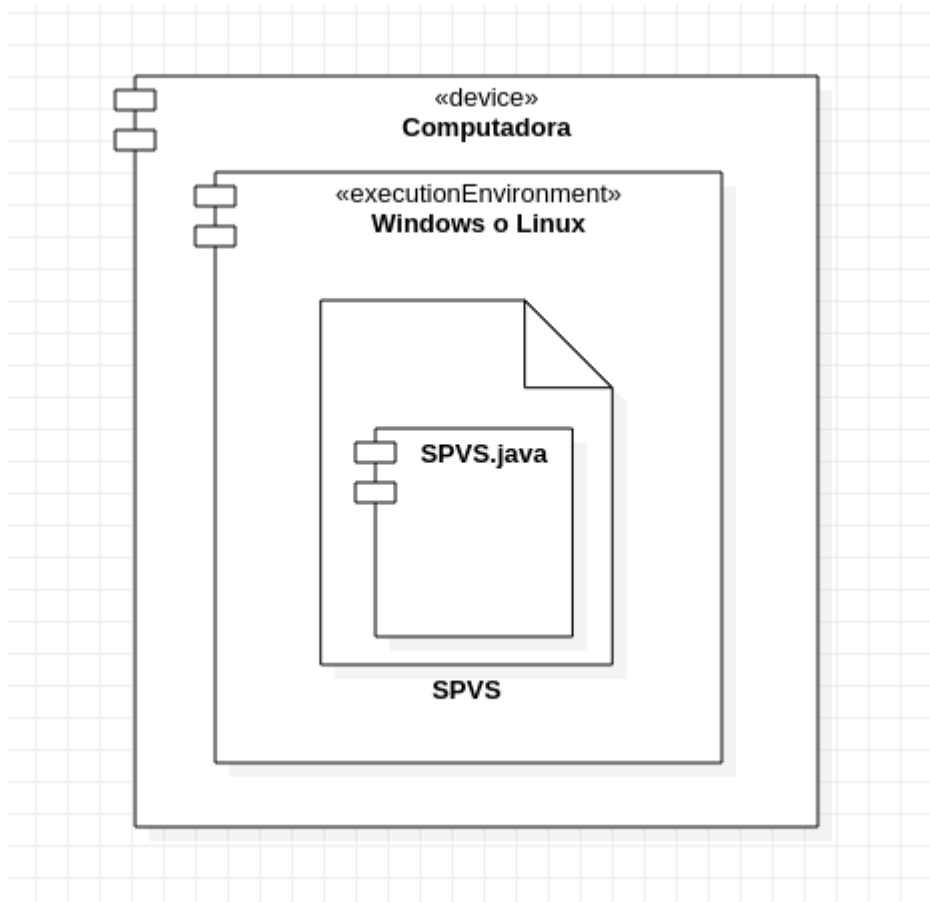
Este patrón de arquitectura nos sirve en nuestros requerimientos no funcionales de la siguiente manera:

Al separar los distintos componentes en módulos apartes genera que sus operaciones críticas estén bien definidas en el lugar donde se realizan, generando que en caso de fallar alguna sea fácil de modificar su error, y optimizar mejor su tarea.

Ésto nos ayudara con nuestros requerimientos no funcionales para generar un mayor rendimiento, confiabilidad en el sistema, y poder mejorar su accesibilidad a partir de sus vistas.

## 11.- Diagrama de Despliegue:

Observamos el Diagrama de Despliegue donde detallamos que nuestro programa ejecutable, se correrá en una computadora con un sistema operativo que puede ser Windows o Linux.



## **12.- Bibliografía:**

Software Engineering, 9th Edition – Ian Sommerville



### 13.- Historial de Versiones:

Nombre	Fecha	Motivo	Versión
Blanco, Murillo	18-06-19	Creación del Documento	1.0.0
Blanco, Murillo	1-06-19	Modificación de Req. No Funcionales Definición de Smoke Tests Desarrollo de Diagrama de Actividades	1.0.1
Blanco, Murillo	15-06-19	Incorporación de Diagrama de Arquitectura, y Despligue Definición del patrón de arquitectura utilizado	1.1.0