

Ingeniería de Software

Trabajo Práctico 2

Documento de Arquitectura

Alumnos:
Blanco, Lucas
Murillo, Sebastian

Docente:
Miceli, Martin

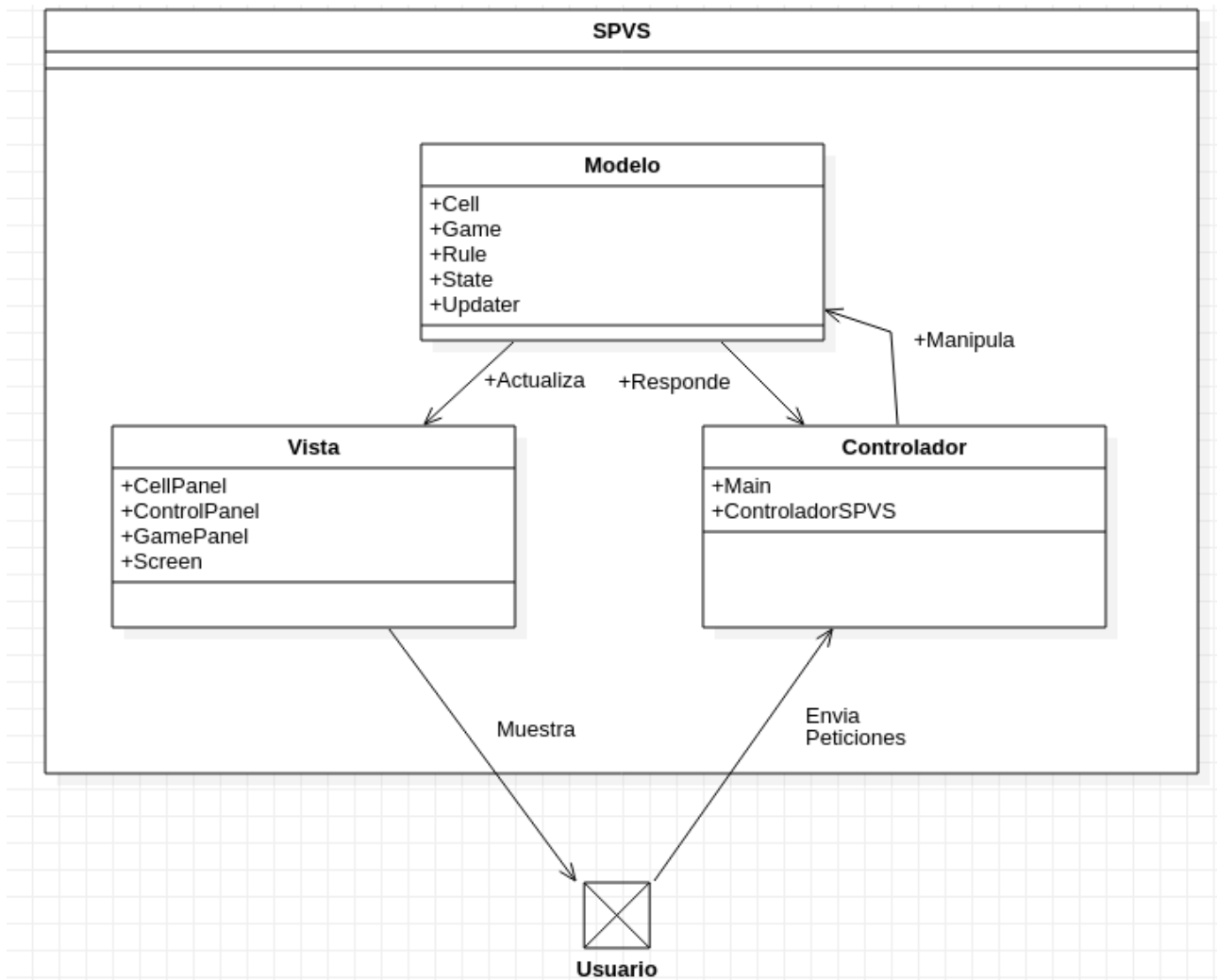
FCEFYN – UNC
2019

Índice:

Diagrama de Arquitectura de Sistema -----	Pág. 3
Patrón de Arquitectura de Sistema -----	Pág. 3
Diagrama de Despliegue -----	Pág. 4
Diagrama de Componentes -----	Pág. 4
Pruebas de Integración -----	Pág. 5
Bibliografía -----	Pág. 6
Historial de Versiones -----	Pág. 7

1.- Diagrama de Arquitectura del Sistema:

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



1.1.- Patrón de Arquitectura:

La arquitectura del programa esta basada en el modelo MVC (Model View Controller), el cuál es un patrón de arquitectura de software, que separa los datos y la lógica de una aplicación de la interfaz de usuario y el modulo encargado de gsetionar los eventos y las comunicaciones.

Para ello MVC propone la construcción de tres componentes distintos que son el modelo, la vista, y el controlador, es decir, por un lado define componentes para la representación de la información, y por otro lado para la interacción del usuario.

En el diagrama de arquitectura anterior se pueden visualizar estas tres partes.

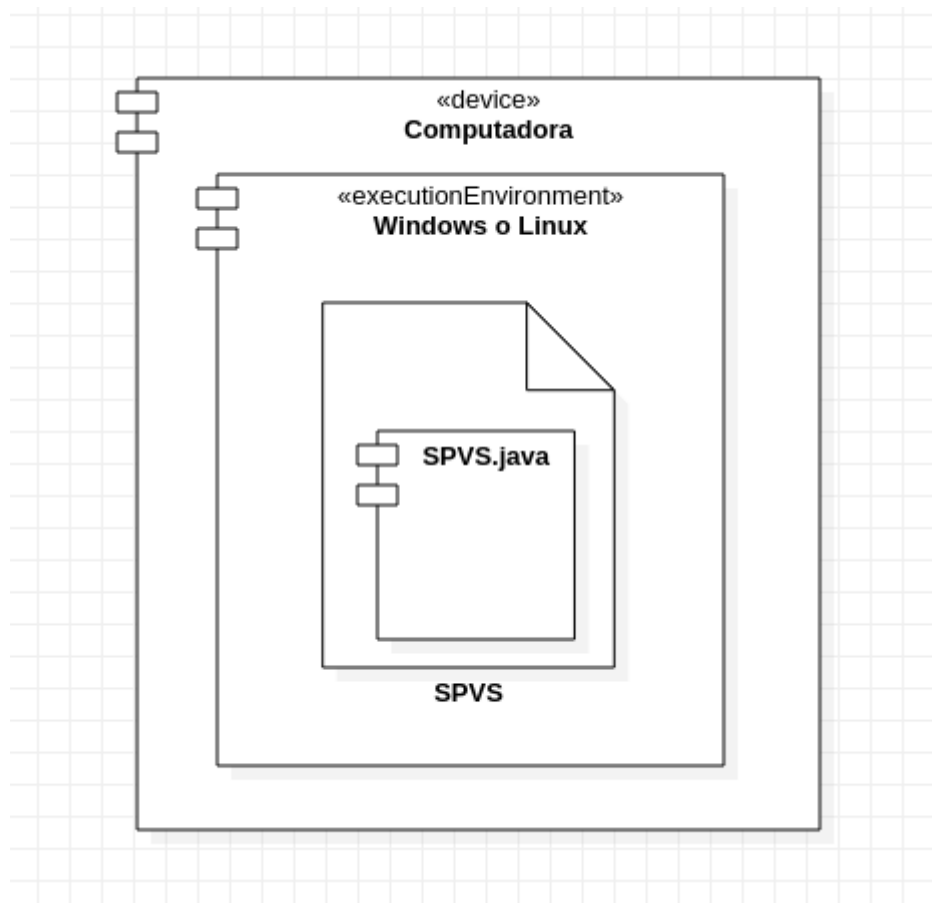
Con este modelo podemos superar requerimientos no funcionales como:

- **Facilidad de uso:** pudiendo dar una interfaz grafica donde el usuario pueda comunicarse directamente con el software.
- **Mantenibilidad:** ya que estando el cogido sectorizado es más fácil encontrar errores.

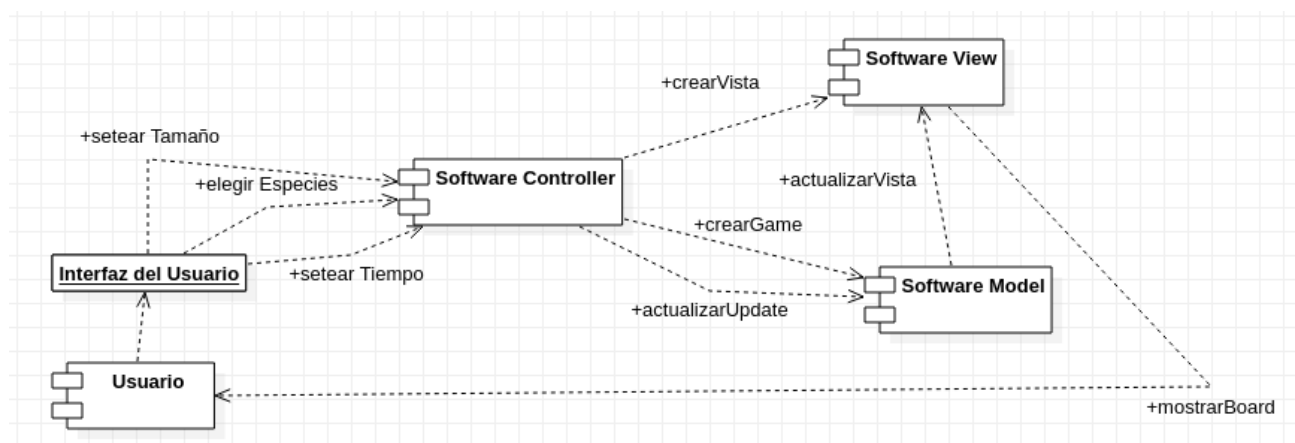
2.- Diagramas:

2.1.- 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.



2.1.- Diagrama de Componentes:



El usuario a través de la Vista se comunicara con el Controlador del software quien actualizará los datos en el Model y estos se mostrarán actualizados en la vista que ve éste.

3.- Pruebas de Integración:

1.- Identificación:	Plan Prueba Integración 001.
2.- Elementos a Probar:	Carga de datos, update de models, update de vista. Finalización a tiempo.
3.- Enfoque:	Ascendente.
4.- Criterio de Aceptación:	Se cargaran los datos correctamente, se actualizara el board con su correspondiente estado, y se finalizará a tiempo.
5.- Criterio de Suspensión:	Cubrir todo el código o fallar en algún elemento.
6.- Tareas:	Preparar código, y mensajes en pantalla si falla alguna parte. Preparar caso de prueba, utilizar la aplicación como usuario final. Evaluar resultados, documentar fallos.
7.- Caso de Prueba:	Abrir aplicación. Esperar para cargar datos. Cargar tamaño board, elegir especies, y tiempo. Ver simulación. Cuando termine, cerrar la aplicación.
8.- Necesidad Ambiental:	Computadora corriendo la aplicación.
9.- Personal:	Persona que realice las pruebas.

Es de gran utilidad también utilizar este mismo plan de prueba pero para prueba negativa, no correspondería netamente con una prueba de integración pero si nos asegurara que el programa esta tratando los errores y no los deja pasar ya que esto nos podría dar errores en la prueba de integración que no estamos teniendo en cuenta. Con negativa nos referimos a forzar al fallo de cada bloque critico.

4.- Bibliografía:

Software Engineering, 9th Edition – Ian Sommerville

5.- Historial de Versiones:

Nombre	Fecha	Motivo	Versión
Blanco, Murillo	15-06-19	Creación del Documento	1.0.0
Blanco, Murillo	26-06-19	Desarrollo de Diagrama de Componentes Desarrollo de Pruebas de Integración	1.1.0