Documento de Arquitectura

[Escribir el subtítulo del documento]

[Seleccionar fecha]

[Escribir el nombre de la compañía]

Benítez Jeremías, Rao Maximiliano, Monsierra Lucas.

Índice.

1. Introducción.
   1. Propósito.
   2. Alcance.
   3. Definiciones y Abreviaturas.
   4. Referencias.
2. Arquitectura del Sistema.
   1. Vista.
   2. Patrón del sistema.
      1. Vista.
      2. Controlador.
      3. Modelo.
3. Vista Física.
4. Vista Lógica.
5. Vista de Despliegue.
6. Vista de Procesos.
7. Vista de Escenarios.

Historial de revisiones.

|  |  |  |  |
| --- | --- | --- | --- |
| Versión | Fecha | Observaciones | Autor |
| 1.0.0 | 26/05/17 | Creación del documento. | Monsierra, Lucas . |
| 1.0.1 | 27/05/17 | Primer versión del documento.  Indice.  Incisos 1 a 2.2. | Benítez, Jeremías |

1. Introducción

1.1 Propósito

Este documento contiene una visión general del Sistema de Reserva de aulas – FCEFYN (de ahora en más “Sistema”). El propósito principal de este Sistema es mejorar la convivencia en el ámbito de la institución, así como también simplificar procesos administrativos permitiendo esto que tanto docentes como no docentes eficienticen las tareas que desempeñen en la facultad.

El fin de este documento es expresar las decisiones arquitecturales importantes que se tomaron durante los procesos de diseño y planificación del Sistema. Sirve también para que las personas encargadas de diseñar, planear e implementarlo puedan entender mejor el enfoque que se espera que tomen a la hora de realizar sus tareas.

1.2 Alcance

El alcance de este documento incluye al Sistema en su totalidad. Esto es, la descripción de sus interfaces gráficas, bases de datos, etc. así también como de las interfaces que permitan la comunicación entre los distintos componentes que forman parte de este proyecto.

1.3 Definiciones y Abreviaturas

// Editar después de terminar el documento.

1.4 Referencias

Sommerville, I., (2011), Ingeniería de software, México, Pearson.

Freeman, E. et al, (2004) Head First Design Patterns, EEUU, O'Reilly

2. Arquitectura del sistema

La arquitectura del Sistema fue diseña para cumplir con determinados objetivos, a saber:

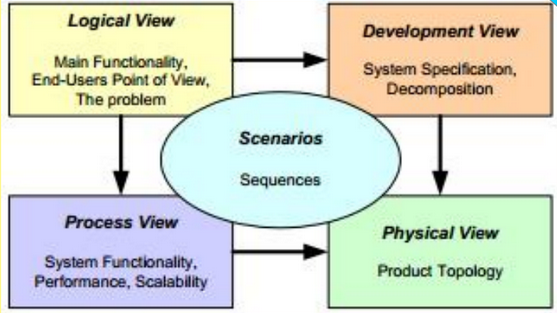
* Cumplir con los requisitos presentados por la materia en la que la aplicación fue inicialmente pensada y diseñada.
* Facilitar la futura modificación. Para esto se buscó que el sistema fuese modular y escalable de forma que la inclusión de futuras mejoras no representara un problema para los equipos involucrados.
* Brindarle mayor flexibilidad al Sistema, permitiendo así que en un futuro el mismo pase a formar parte de otro sistema de mayor alcance.

2.1 Vista

Cualquier sistema que se desee implementar necesita ser representado desde distintos puntos de vista. Al igual que para construir una casa se necesita proveer a las partes que van a estar implicadas en su construcción diversos planos que representen la misma vivienda, vista desde distintos aspectos (planos del tendido eléctrico, planos de la estructura en sí, planos de las cañerías, etc.); un sistema de software debe representarse desde distintos puntos de vista.

Esto es así para ofrecer una representación más sólida de lo que se desea implementar, evitando que existan vaguedades o ambigüedades en la descripción del mismo. Uno de los beneficios de actuar de esta forma es que se disminuye la probabilidad de que un desarrollador deba tener que tomar por su cuenta decisiones de diseño, las cuales deberían ser tomadas por el equipo correspondiente. Esto lleva, en general, a sistemas más consistentes.

Teniendo en cuenta esto, se optó por elegir la vista 4+1 para representar esta aplicación. La misma cuenta, como su nombre infiere, con cuatro vistas (lógica, física, de despliegue y de procesos) más una que las relaciona (de escenarios).

Diagrama básico de la vista 4+1

Se incluye a continuación, una breve descripción de qué se debe esperar ver representado en cada una de las mismas:

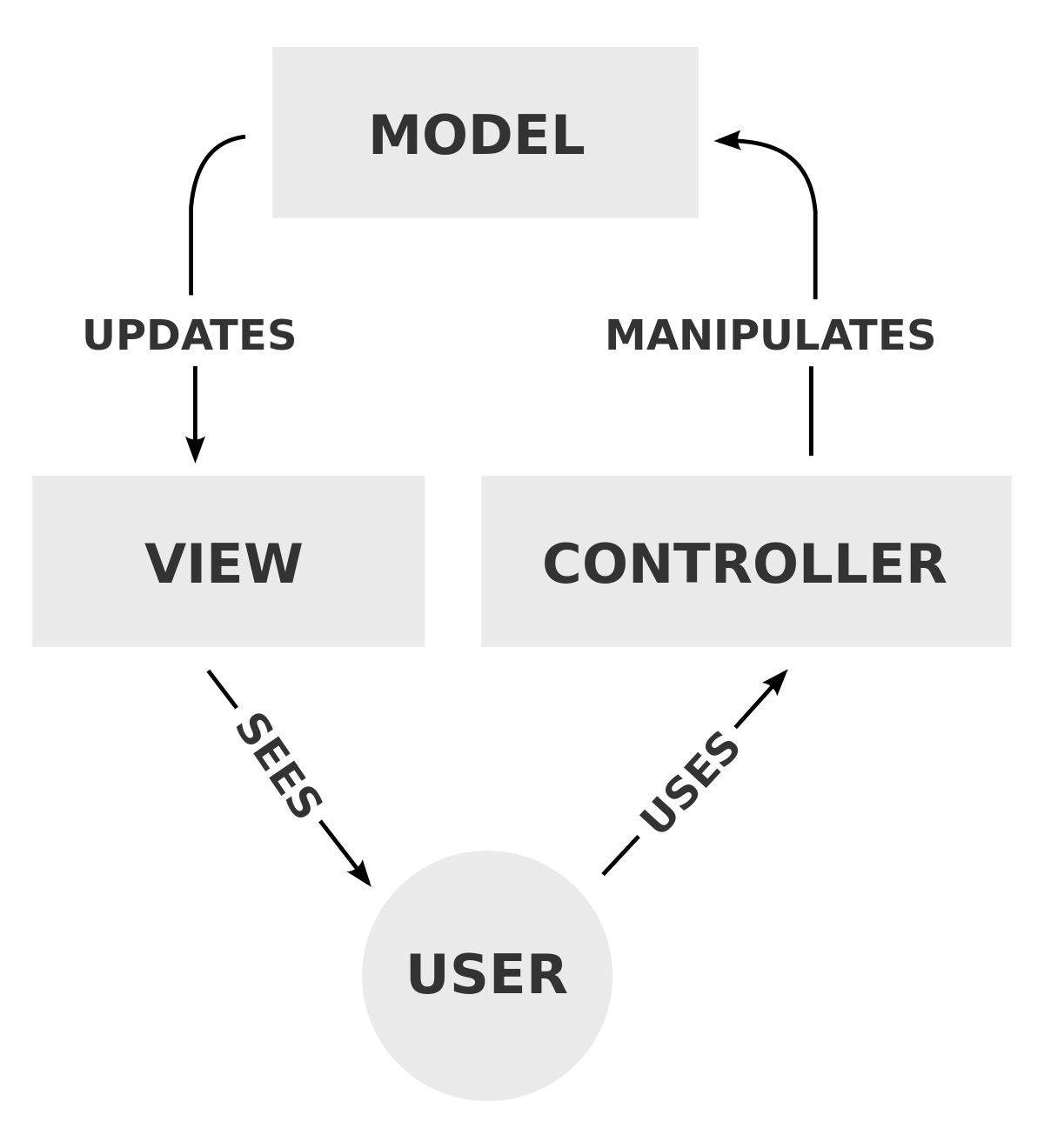
* Vista Lógica: Esta se encarga de representar la funcionalidad principal de los usuarios finales. En este proyecto, la misma incluye // AGREGAR CUALES INCLUYE
* Vista Física: Es la que representa los componentes físicos del sistema, así como la conexión (física también) que hay entre ellos. Está dada para este caso como un Diagrama de despliegue.
* Vista de Despliegue: Esta es a la que deberán recurrir los programadores y todo aquel que desee conocer cómo está dividido el software y las dependencias que hay entre sus componentes. // AGREGAR CUALES INCLUYE
* Vista de Procesos: Aquí se representan los procesos que hay en el sistema y la forma que tienen los mismos para comunicarse. // AGREGAR CUALES INCLUYE
* Vista de Escenarios: Es la que “une” las demás vistas mediante casos de uso con lo que tendremos una trazabilidad de componentes, clases, equipos, paquetes, etc., para realizar cada caso de uso. En este documento se puede ver como diagramas de casos de uso.

2.2 Patrón del sistema

Debido a los requerimientos de los que se habló en la introducción a esta sección, fue que se optó por elegir un patrón de diseño conocido como MVC (model-view-controller).

La consecuencia de esta elección es que las vistas (en este caso las interfaces gráficas que serán brindadas al usuario), los controladores (que serán las porciones de código que se encarguen de manejar el comportamiento del sistema, sirviendo como intermediario si se quiere entre las vistas y el modelo) y el modelo (la abstracción del problema en sí, en este caso las bases de datos y las operaciones que se puedan realizar sobre ellas) estén lo más separadas posibles.

Así, el proceso de implementar cambios en cualquiera de estos tres componentes se ve drásticamente simplificado, siempre que se respeten las interfaces entre los tres.

Diagrama básico del patrón MVC

2.2.1 Vista

2.2.2 Controlador

2.2.3 Modelo