Documento de requisitos

Del

*“Sistema para el control de asistencia escolar AE 1.0”*



|  |  |
| --- | --- |
| Fecha | 19/05/2015 |
| Versión | 1.0 |
| Integrantes | Amairany Ramos Meraz. |
| Del grupo | David Arcadio Arredondo Sainz. |
|  | José Agustín Esquerra Morales. |
|  | Jorge Eduardo Silva Carrillo. |
|  | Claudia Janeth Román Navarro.  Joaquín García Félix |

Prefacio

Este es el documento de requisitos del proyecto de un sistema Web para el control de Asistencia Escolar de alumnos de la UNIDEP llamado "AE 1.0", el cual tiene como propósito tener un control de las asistencias en las escuelas (registro de asistencia de los alumnos, reportes, etc.).

**Alcance del documento**

El documento de requisitos es la base de todo el desarrollo futuro de *“AE 1.0”*. Describe los siguientes aspectos del sistema:

Propósito, contexto, requisitos funcionales, requisitos de pruebas, requisitos de ambiente, arquitectura del sistema y riesgos del proyecto.

**Documentos relacionados**

Documento de inicio de proyecto, proyecto “AE 1.0".

**Autores**

Amairany Ramos Meraz, David Arcadio Arredondo Sainz, José Agustín Esquerra Morales, Jorge Eduardo Silva Carrillo, Claudia Janeth Román Navarro, Joaquín García Félix.

**Lectores**

Este documento está dirigido principalmente a los desarrolladores del proyecto, como también es de utilidad para todos los interesados en el mismo.

|  |  |  |  |
| --- | --- | --- | --- |
| Versión | Fecha | Explicación del cambio | Autor |
| 1.0 | 19/05/2015 | Primer borrador | Amairany Ramos Meraz, David Arcadio Arredondo Sainz. |
| 1.1 | 25/05/2015 | Edición de introducción | Amairany Ramos Meraz |
| 1.2 | 25/05/2015 | Creación del logotipo del sistema. | José Agustín Esquerra Morales |

1. **Introducción**

En esta introducción se describe brevemente el contexto, objetivos y alcance del proyecto a desarrollar, así como la documentación relativa al mismo. Esta información está basada en el documento de inicio de proyecto.

1. Propósito

El proyecto a desarrollar busca automatizar los procesos de las asistencias en el salón de clases, para que todo el proceso se realice de forma electrónica por medio de una computadora o Smartphone que tenga servicio a internet.

2. Alcance

El proyecto está dirigido para la UNIDEP, hasta llegar a la fase de implementación.

3. Contexto

El programa que se desarrolla se está elaborando en el framework Laravel y se trabaja con el editor de textos Sublime Text 2.

Desarrollaremos todo basándonos en las necesidades genéricas de los maestros de las escuelas.

4. Referencias

**2.- Requisitos del sistema**

* 1. ***Requisitos Funcionales***

RF1.- La aplicación debe permitir calendarizar el apartado de laboratorios.

RF2.- La aplicación debe poder dar de alta nuevos artículos de laboratorio.

RF3.- La aplicación debe poder registrar la entrada de los alumnos al laboratorio.

RF4.- Debe poder registrar que materiales le fueron prestados a cada alumno.

RF5.- Se debe poder visualizar las prácticas filtradas por:

\*Maestro

\*Materia

RF6.- La aplicación debe ser capaz de facilitar la ubicación de los artículos, de acuerdo al estante en el que se encuentra en el almacén.

RF7.- La aplicación debe de mostrar solo los artículos que se encuentran en el laboratorio que se requiere.

RF8.- Debe de registrar los artículos que el alumno entrega al finalizar su práctica.

RF9.- Se debe poder marcar a un alumno con adeudo, en caso de que este no entregue el material completo.

RF10.- Se debe agregar automáticamente la actividad de práctica en Moodle para que el alumno pueda subir el reporte de su práctica.

RF11.- El encargado de laboratorio debe de recibir las solicitudes que el maestro envía para calendarizar una práctica.

RF12.- El encargado de laboratorio debe de autorizar el uso del laboratorio en la fecha que el maestro ha solicitado.

RF13.- El maestro debe de confirmar que se llevará a cabo la práctica mínimo con una semana de anticipación.

RF14.- El encargado de laboratorio y el jefe de departamento deben de poder generar los reportes necesarios.

***2.2.- Requisitos no funcionales***

RNF1.- Se debe poder ingresar a la aplicación en todo momento.

RNF2.- La creación de reportes debe estar actualizada en todo momento.

RNF3.- Los datos de los usuarios deben encontrarse protegidos.

RNF4.- Se debe poder ingresar a la aplicación rápidamente.

RNF5.- La aplicación debe contener una interfaz intuitiva (fácil de usar).

RNF6.- La aplicación deberá funcionar en cualquier sistema operativo y cualquier dispositivo, siempre y cuando este cuente con un explorador como (Google Chrome, Mozilla Firefox).

RNF7.- La aplicación debe ser estable.

RNF8.- El sistema basa su seguridad en la autenticación de usuarios, donde se le asigna a cada usuario a excepción de los alumnos, un usuario y contraseña, para que este pueda acceder a la aplicación.

RNF9.- La aplicación debe funcionar de forma correcta.

RNF10.- La aplicación debe tener la capacidad de agregar nuevas funcionalidades o módulos de una forma sencilla.

RNF11.- La aplicación no debe tener ningún conflicto en la interacción con otros sistemas.

RNF12.- La aplicación debe ser de fácil mantenimiento.

RNF13.- La aplicación debe tener un tiempo de respuesta rápido, sin importar la cantidad de usuarios que estén trabajando en ella.

RNF14.- Los datos de la aplicación solo podrán ser modificados por aquellas personas autorizadas para ello.

RNF15.- El sistema debe poder recuperarse rápidamente de un fallo, si es que llegara a suceder.

***2.3.- Requisitos de interfaces.***

La tabla 1 muestra la lista de eventos externos a los que el sistema responde. La primera columna es el nombre del evento; la segunda es la descripción del mismo. El “iniciador” es la componente externa al sistema que inicia el evento. Los parámetros son los datos asociados al evento. La respuesta es el nombre de una respuesta, cuya descripción está en la tabla 2.

Tabla 1: Eventos externos

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Evento | Descripción | Iniciador | Parámetros | Respuestas |
| Evento externo 1 | La aplicación debe mostrar la información cuando un usuario se autentifique | El usuario al ingresar a la aplicación. | El parámetro será el usuario que se debe autentificar con un usuario y contraseña correctos. | Mostrar los datos. |
| Evento externo 2 | La aplicación debe detectar que tipo de usuario es el que está entrando a ella. | El usuario al ingresar a la aplicación | El usuario, que debe autentificarse con un usuario y contraseña válidos. | Mostrar la información correspondiente al tipo de usuario que pertenece. |
| Evento externo 3 | La aplicación debe detectar si se encuentra ocupada la fecha solicitada, para apartar el laboratorio. | El usuario reservando un laboratorio. | Un nombre de laboratorio valido, nombre de maestro, una fecha de reservación, un nombre de práctica. | Si la fecha solicitada está ocupada, o no. |
| Evento externo 4 | La aplicación debe registrar la entrada de alumnos al laboratorio | El alumno al registrar su número de control en la aplicación | El alumno que debe registrarse con un número de control valido y con vigencia. | Un mensaje de registro exitoso. |

La tabla 2 muestra las respuestas del sistema frente a eventos externos.

Tabla 2: Respuestas del sistema

|  |  |  |
| --- | --- | --- |
| Respuesta | Descripción | Parámetros |
| Respuesta al  evento 1 | Se iniciará la aplicación mediante la el valor de entrada de usuario y contraseña | Detectará la modificación de los campos de usuario y contraseña. |
| Respuesta al evento2 | La aplicación detectará el usuario con el que se autentificó y verificará en la base de datos que tipo de usuario es y con qué es lo que debe ver de la aplicación. | Detectará la modificación de los campos de usuario y contraseña. |
| Respuesta al evento 3 | La aplicación detectará si en la fecha, hora solicitada y en el laboratorio indicado, no se encuentra registrada ya una práctica. | El nombre de la práctica, el laboratorio, el horario y la fecha. |
| Respuesta al evento 4 | La aplicación almacenará en una base de datos al alumno que se registró. | Detectará el número de control del alumno. |

***2.4.- Actores del sistema y roles.***

|  |  |
| --- | --- |
| Actores | Roles |
| Alumnos | 1.- Registrar su asistencia a la práctica |
| Maestros | 1.- Solicitar la práctica 2.- Confirmar la asistencia a la práctica. 3.- Dar de alta prácticas.  4.- Dar de baja prácticas. |
| Jefes de departamento | 1.- Ver reportes de la actividad de las prácticas. |
| Jefes de laboratorio | 1.- Dar de alta/ baja prácticas. 2.- Dar de alta o baja/ artículos 3.- Ver reporte de la actividad de las prácticas. 4.- Rechazar o admitir la práctica que solicite un maestro. 5.- Registra qué material fue prestado a que alumno o maestro. 6.- Registra la devolución del material prestado.  7.- Genera un adeudo de material, si este no fue entregado en su totalidad o se dañó. |
| Auxiliares de laboratorio | 1.- Registrar material que fue prestado a que alumno o maestro.  2.- Registra la devolución del material prestado.  3.- Genera un adeudo de material, si este no fue entregado en su totalidad o se dañó. |

***2.5.- Casos de uso. (3 o 4 casos de uso)***

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Caso de uso | Registro de una práctica |  | CU1 |  |  |
| Actores | Maestro, jefe del laboratorio |  |  |  |  |
| Tipo | Primario |  |  |  |  |
| Referencias | El maestro debe estar registrado en el sistema, tener una materia asignada |  |  |  |  |
| Precondición | El sistema debe estar funcionando correctamente. |  |  |  |  |
| Poscondición | La aplicación ya no podrá mostrar libre la fecha ya solicitada por un maestro para cierto laboratorio |  |  |  |  |
| Autor |  | Fecha |  | Versión | 1.0 |
|  |  |  |  |  |  |

|  |
| --- |
| Propósito |
| Registrar una práctica de un laboratorio determinado. |

|  |
| --- |
| Resumen |
| Un maestro se autentifica en la aplicación, selecciona que desea registrar una práctica, especifica el nombre de la materia y si la práctica ya se encuentra registrada la elige, selecciona la cantidad de alumnos y la fecha en la que desea realizarla. |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Caso de uso | Registro de asistencia a práctica |  | CU2 |  |  |
| Actores | Alumnos |  |  |  |  |
| Tipo | Primario |  |  |  |  |
| Referencias | El alumno debe tener una práctica asignada en esa fecha. | CU2 |  |  |  |
| Precondición | El sistema debe estar funcionando correctamente. |  |  |  |  |
| Poscondición | Se registrará en la base de datos la asistencia del alumno a dicha práctica en la fecha actual |  |  |  |  |
| Autor |  | Fecha |  | Versión | 1.0 |
|  |  |  |  |  |  |

|  |
| --- |
| Propósito |
| Registrar la entrada del alumno a una práctica. |

|  |
| --- |
| Resumen |
| Un alumno registra su número de control, el nombre del maestro y la práctica a la que asiste. |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Caso de uso | Registro de préstamo de material de laboratorio |  | CU3 |  |  |
| Actores | Maestro, jefe del laboratorio, alumno |  |  |  |  |
| Tipo | Primario |  |  |  |  |
| Referencias | El solicitante del material o equipo debe estar registrado en el sistema, tener una práctica asignada ese día y a esa hora. | CU2 |  |  |  |
| Precondición | Deberá verificar que se cuente con el material que se solicita. |  |  |  |  |
| Poscondición | La aplicación modificará la cantidad de material o equipo disponible. |  |  |  |  |
| Autor |  | Fecha |  | Versión | 1.0 |
|  |  |  |  |  |  |

|  |
| --- |
| Propósito |
| Registrar el préstamo de material o equipo de laboratorio. |

|  |
| --- |
| Resumen |
| Un alumno o maestro solicita al encargado o auxiliar de laboratorio ciertos materiales y/o equipo, este debe registrar la salida de ese material y generar un adeudo de material. |

**3.- Arquitectura del sistema**

***3.1.- Definir arquitectura***

La arquitectura que creemos que es la mejor para desarrollar nuestra aplicación es: una arquitectura cliente servidor con un patrón de arquitectura modelo vista controlador.

Arquitectura cliente-servidor



Patron modelo vista controlador (MVC)



***3.2.- ¿Por qué se eligió esa arquitectura?***

La arquitectura cliente/servidor permite el mejor aprovechamiento de los recursos de cómputo disponibles. Una buena implementación de la arquitectura, reduce de manera considerable el tráfico en la red. Permite el uso de interfaces gráficas variadas y versátiles, lo que conlleva a interfaces más dinámicas y amigables con el usuario. Este tipo de arquitectura, facilita la integración entre sistemas diferentes.

***3.3.- Descripción detallada de la arquitectura (Identificar los elementos o componentes de la arquitectura)***

Capa de Presentación (Cliente)

El lenguaje a utilizar para la presentación es HTML y CSS.

|  |  |
| --- | --- |
| Nombre de Interfaz | Descripción |
| Pantalla de autentificación | En esta pantalla es la que se utilizará para que los usuarios registrados tengan acceso a la aplicación. |
| Pantalla de solicitud de laboratorio | En esta pantalla se podrá realizar una solicitud para hacer un apartado de un laboratorio específico a una fecha específica dando el motivo de por qué desea realizarse la práctica y el periodo en el cual se desea realizar. |
| Pantalla para visualizar solicitudes | En esta pantalla se podrá visualizar las solicitudes hechas a un laboratorio determinado. |
| Pantalla para visualizar prácticas calendarizadas | Esta pantalla se podrán visualizar las prácticas que ya se encuentran calendarizadas, que día se llevará a cabo, que maestro dará la práctica, quien es el responsable de realizar la práctica, comentarios y el periodo en el que se realizará. |
| Pantalla registro de alumnos. | En esta pantalla el alumno ingresará su número de control para registrar su entrada de laboratorio, seleccionará la práctica a la que asiste, el maestro, la materia y la hora de entrada se tomará del sistema. |
| Pantalla de préstamo de artículos | En esta pantalla se registrarán los artículos y la cantidad que el alumno necesitará. |
| Pantalla de inventario | En esta pantalla se podrá dar de alta o baja un artículo, indicar si el artículo se encuentra en reparación y si tiene alguna merma.  También se podrá ver la cantidad que hay de un artículo específico así como sus datos. |
| Pantalla de reportes | En esta pantalla se podrán consultar los distintos reportes como:   * Cantidad de alumnos atendidos en algún periodo. * Artículos que han tenido mayor o menor movimiento. * Adeudos de materiales. * Cantidad de prácticas realizadas en un periodo especifico.   Ente otros. |
| … | .. |
| … | .. |

Capa de procesamiento (Lógica de negocio).-

El lenguaje a utilizar para la lógica de negocio será Laravel.

Controladores:

* Laboratorios.
* Alumnos.
* Artículos.
* Responsables.
* Prácticas.

Modelos

* Laboratorios.
* Alumnos.
* Artículos.
* Responsables.
* Prácticas.

Archivos de rutas

Capa de gestión de datos.-

Para la gestión de los datos utilizaremos el sistema gestor de base de datos MySql.



***3.4.- Asignar roles a los miembros del equipo de trabajo.***

|  |  |  |
| --- | --- | --- |
| Nombre | Rol | Descripción |
| Jimenez Iñiguez Edwin Fernando. | Frontend. | Persona que diseñara las interfaces. |
| López Valenzuela Ana Patricia. | Frontend. | Persona que diseñara las interfaces. |
| Meza Duarte Othoniel. | Arquitecto de software. | Define la arquitectura |
| Moreno Zazueta Judith Gpe. | Capa de gestión de datos. | Diseña el modelo de datos y crea la base de datos. |
| Silman Montenegro Rubén. | Backend. | Se encarga de darle sentido a la lógica del negocio. |