Intranet para la administración de un Colegio:

Sistema de administración para un colegio.

EL Sistema permite un rápido manejo de la información que necesita un colegio, como:

Detalles de alumno.

Detalles de Docentes.

Detalles de Administradores

**Requerimientos del Sistema:**

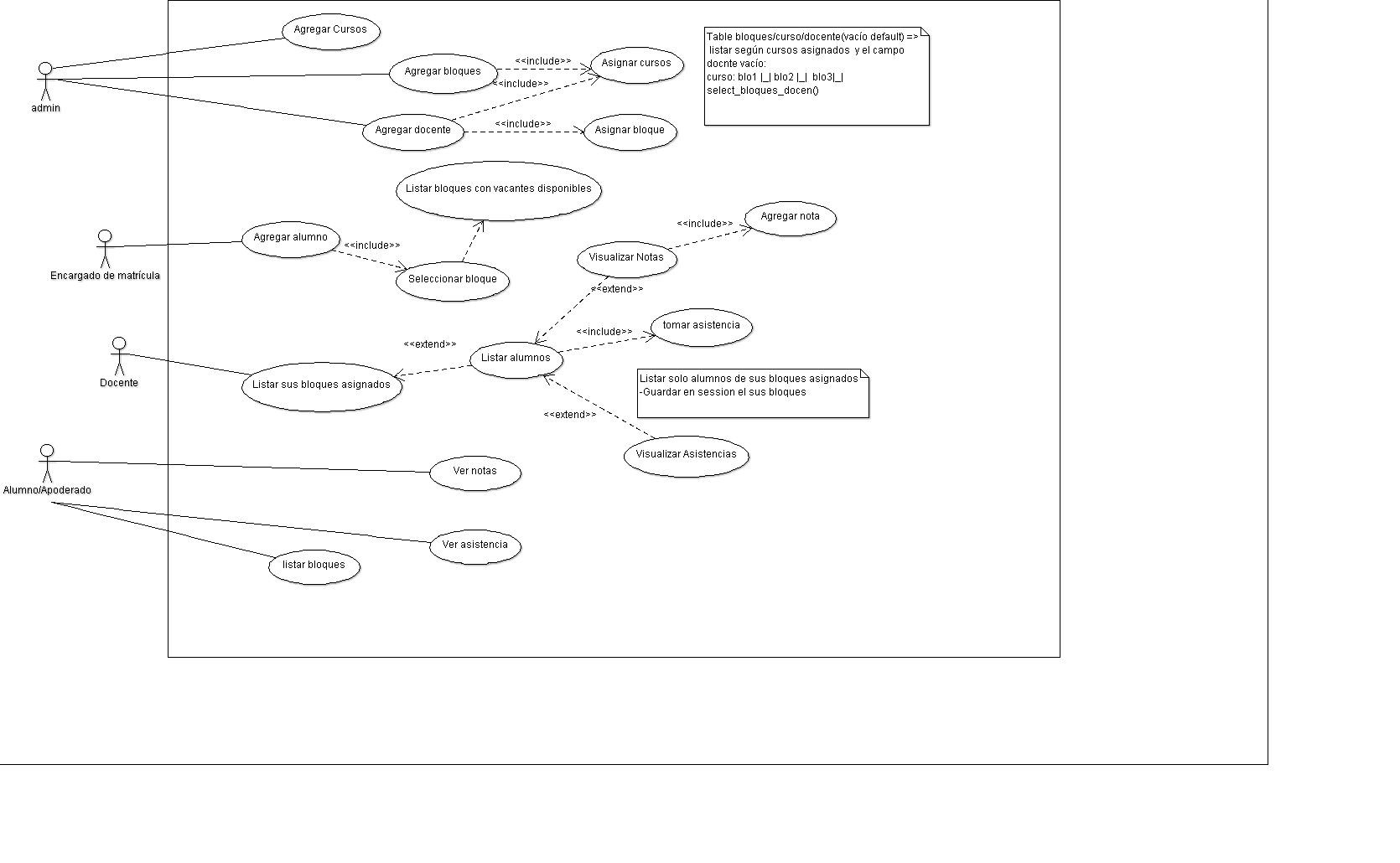
-Poder ingresar asistencia con total libertad, tener en cuenta que cada curso toma su propia asistencia.

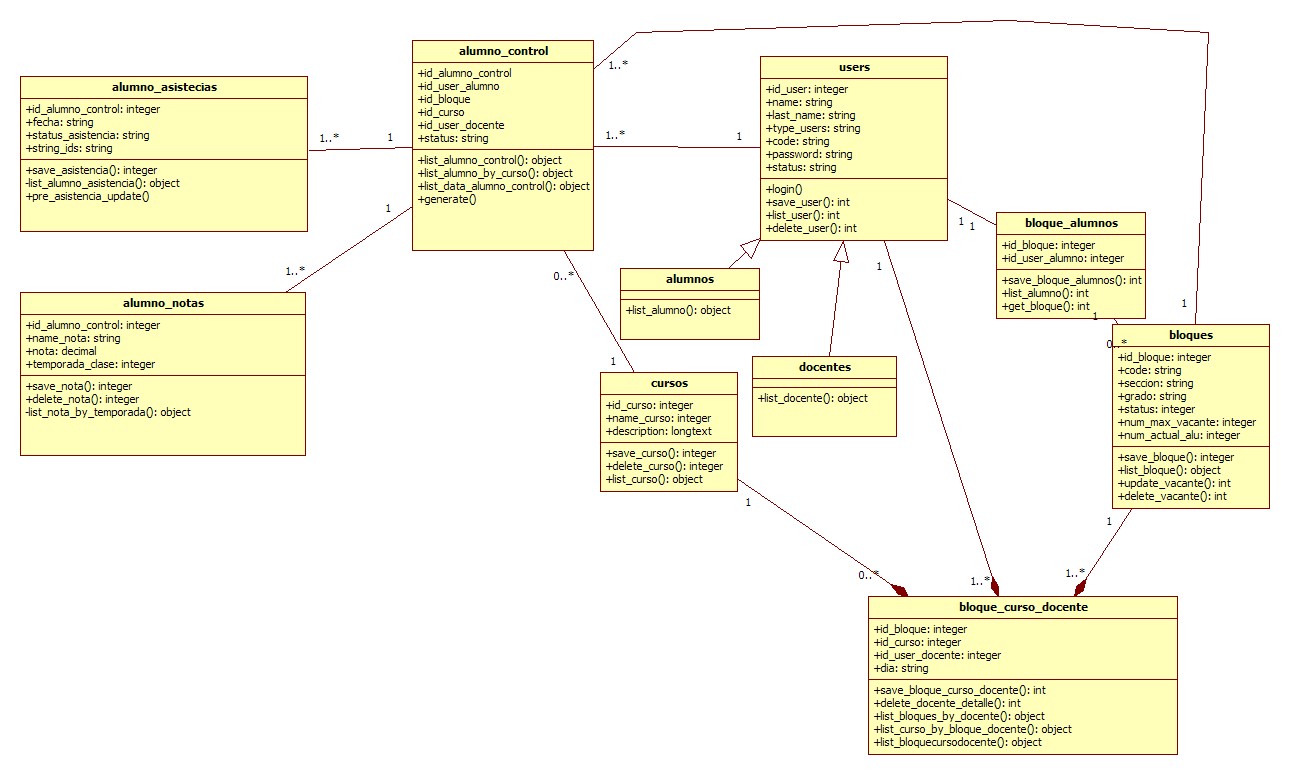
-Poder ingresar notas de un alumno.

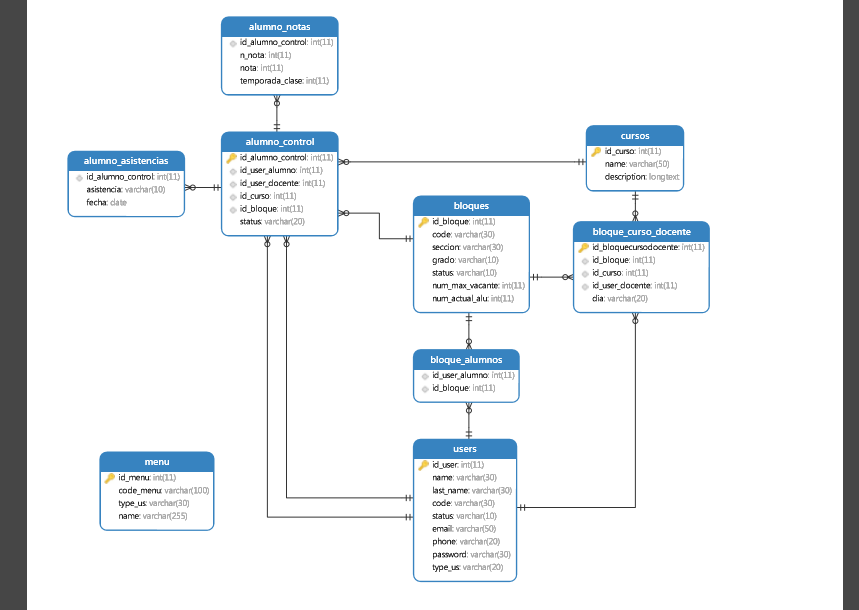
-Listar informe total del alumno para ser visto por sus apoderados.

-Administración de usuarios.

Desarrollo del sistema:

Creación de los casos de uso y actores que interactúan con el sistema:

Creación de diagrama de clases:

Diagrama de la base de datos:

**Uso:**

Para comenzar, el sistema se crea con un user por defecto de forma manual (user ROOT) y a partir de ese usuario podrá crear los diferentes tipos.

Tipos de Usuarios:

**ROOT:**

Encargado de deshacer eliminaciones de usuarios y la creación del mismo, por defecto solo se usa para crear a un admin.

**ADMIN:**

encargado del ingreso de alumnos y docentes. Asigna a los docentes a un curso y a un bloque según las capacidades del maestro.

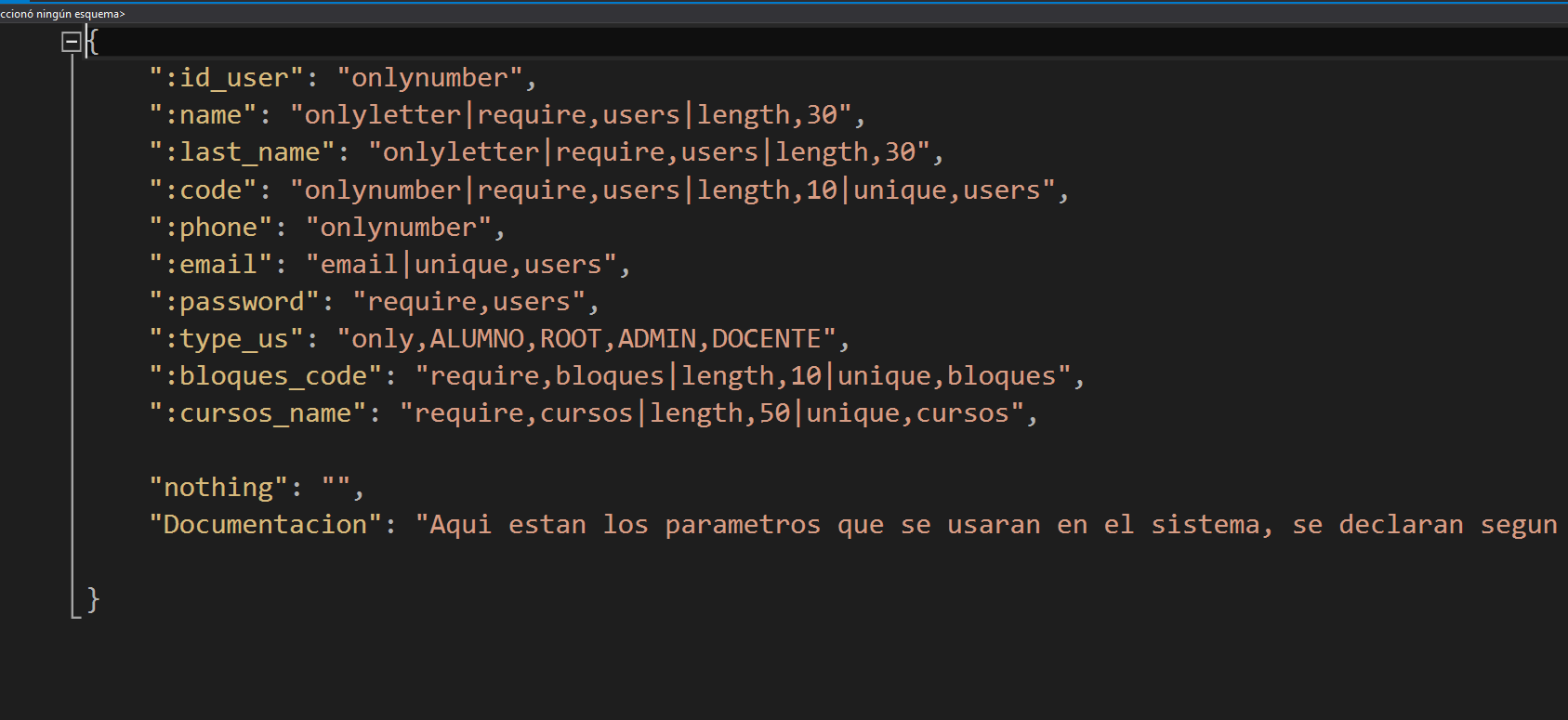
Creación y administración de bloques y cursos.

**DOCENTE:**

Encargado de las notas de los alumnos y su asistencia a su clase.

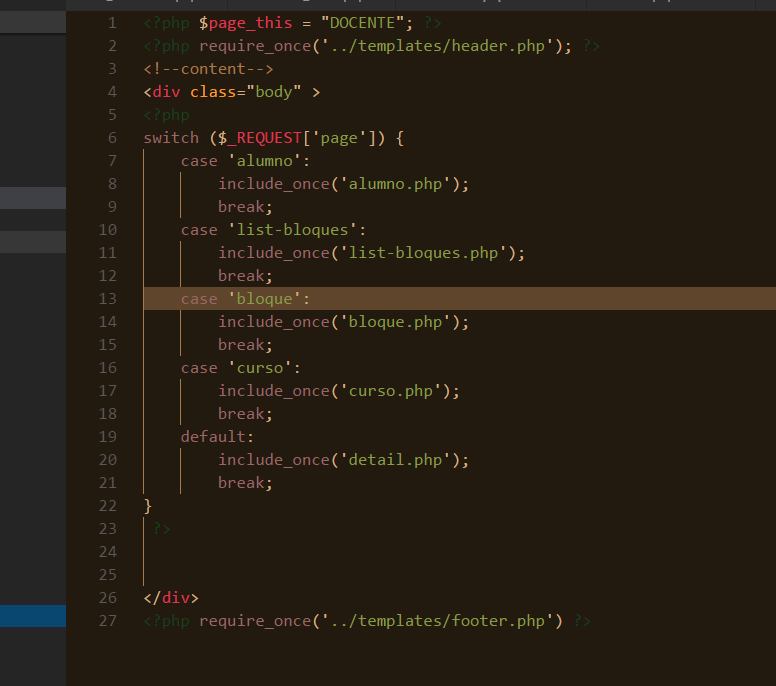
**ALUMNO:**

Listado de su información que tiene el colegio del alumno. Listado de su asistencia y sus notas dependiendo de la temporada de clases.

EL ingreso de cualquier usuario está validado por un archivo json para que sea fácil su manipulación: \www.school\_stark.dev\controllers/params\_validate.josn

En el archivo controllers/defines.php proporciona variables globales que puedes ser de utilidad al sistema.

El sistema posee un control de página, es decir que no permite navegar por páginas que no se han especificado para el usuario, esto es gracias al uso de .htaccess y una página master que se usa de control para las demás páginas (la página es el ‘Index.php’ en cada módulo)



El uso de ‘require\_once’ permite que solo se llame a un archivo una sola vez así se evita llenar código de más a la página.

**Sobre la calidad del sistema:**

**Funcionalidad:**

El sistema es estable y maneja el tema de seguridad sin ningún problema, con las validaciones de tipo de usuario y la redirección de usuario no permitido.

Abarca los requisitos de listado y reporte que se han solicitado.

Permite el ingreso de usuarios con las validaciones correspondientes. Y no permite que haya ambigüedad con los usuarios.

**Fiabilidad:**

Como ya se mencionó el sistema tiene una validación por tipo de usuarios y no permitirá que un usuario no deseado entre.

Usa PDO para hacer las consultas a la base de datos e ingresa datos por parámetros de esta clase dada por PHP esto permite que no sea posible ataques por inyección sql. ***"'; DELETE FROM users; /\*"***

También al ser PDO soporta otras base de datos aparte de MySQL, así que se puede migrar sin dificultad.

La eliminación solo se da por el campo status, es decir ningún usuario se elimina solo se desactiva, así podrá ser simple su recuperación.

**Usabilidad:**

El sistema usa un paquete de estilos llamado ‘Pure’ que es simple y de diseño amigable y moderno.

Con respecto al código, el sistema está dividido en procesos (clases) y las vistas de la misma forma, así podrá ser rápida la implementación de módulos en el futuro y no perjudica a todo el sistema.

Las URL se manejan por .htaccess, esto hace más amigable las url para la vista del cliente.

**Eficiencia:**

El sistema usa css y js comprimidos para que la página no tenga demoras al cargar.

Tiene el uso de url amigables para que el usuario entienda el proceso en el que se encuentra.

**Mantenimiento:**

El mantenimiento del sistema se hace con facilidad, debido a que los procesos son independientes entre sí.

EL código se realizó con variables estándares y nombre de funciones que sea legibles por lo demás y la mayoría de las funciones están comentadas con el proceso que cumple.

El diseño de vista está dividido por header, footer, body y nav , así se puede dar un mantenimiento de forma independiente a cada parte sin afectar a la otra.

**Movilidad:**

Al ser un sistema web está optimizado para cualquier ambiente moderno, me refiero a cualquier navegador.

El lenguaje de programación es PHP así que es confiable con la base de datos que sea compatibles con dicho lenguaje.