1-8-2016

José Angel Pereira / José Gregorio Castillo

Spontecorp

Proyecto Colegio Belagua

Documento de diseño del sistema

Contenido:

# Proyecto Colegio Belagua, diseño del sistema:

## Definición del Problema:

La administración del Colegio Belagua (Guatire –edo Miranda), desea contar con una aplicación web que permita a sus usuarios: personal administrativo, profesores y representantes ingresar a este sistema, por medio de una autenticación y una vez allí realizar ciertas tareas.

Los usuarios del sistema deben estar en capacidad de auto-registrarse en la aplicación, la cual será verificada y autorizada.

El sistema debe tener un encargado del mismo (configurador) que permita cargar la información relevante a la carga de:

* Puede crear usuarios directamente.
* Año lectivo,
* Niveles educativos.
* Cantidad de cursos – secciones de ese año.
* Respaldo y Recuperación de información de alumnos y del sistema.

Además del usuario configurador el sistema debe permitir y discriminar otros tipos de usuarios: personal administrativo, personal docente, representantes de alumnos. En sistema no está pensado para que los alumnos puedan ingresar a él.

Personal administrativo: es aquel que realiza labores administrativas en el colegio. El sistema debe permitirle a este usuario poder:

* Cargar información de los pagos de acuerdo al nivel/curso de cada alumno.
* Verificar los pagos recibidos en el sistema por el representante de uno o más alumnos.
* El usuario administrativo está en capacidad de configurar la información pertinente del personal docente del instituto.
* El usuario administrativo puede cargar y editar la información de un alumno. Esta información proviene del sistema administrativo del colegio y será alimentada a este sistema en una forma que se determinará luego, pudiendo ser entre las siguientes opciones:
  + Un archivo de texto (txt)
  + Un archivo Excel
  + De forma manual
* Emitir informes que le permitan ver:
  + El estado de cuenta de un alumno
  + El estado de morosidad de un nivel/curso/sección
  + Un estado de cuenta por:
    - Representante
    - Nivel
    - Curso
    - Sección
    - General
* Envío, desde el sistema, correos electrónicos de diferente manera:
  + a un representante,
  + a un grupo de representantes,
  + a un docente,
  + a un grupo de docentes,
  + a otro(s) usuarios administrativos
  + la combinación de estos usuarios que se requiera
  + a toda la comunidad

Personal docente: es aquel que realiza labores de enseñanza en el colegio. El sistema debe permitirle a este usuario

* Únicamente el envío de correos electrónicos a:
  + Un representante ( o varios)
  + A un (o varios) personal administrativo

Representantes: es toda aquella persona que representa a uno o más alumnos. El sistema debe permitirle a este usuario:

* Ver el estado de cuenta del alumno (o alumnos) que representa
* Notificar pago realizados a la cuenta

Alumnos: en la actual versión no son usuarios del sistema, por lo que no tendrán acceso al mismo. Un alumno puede estar representado por uno o más representantes.

## Lista de Características:

Esta lista representa las tareas que debe realizar el sistema, la numeración no implica ni un orden de ejecución ni de importancia.

1. El sistema debe permitir guardar / editar / recuperar información de la base de datos
2. El sistema debe ser capaz de identificar un usuario y discriminar su funcionalidad de acuerdo a su rol
3. El sistema debe contar con un usuario de tipo configurador.
4. El sistema debe ser capaz de representar a un alumno y sus representantes.
5. El sistema debe ser capaz de asociar a un alumno a un nivel/curso/sección.
6. El sistema debe contar con una forma que permita registrarse e ingresar al mismo a un personal administrativo, a un docente y a un representante.
7. El sistema debe permitir al usuario configurador la carga de información del año lectivo, de la cantidad de niveles, cursos y secciones en cada nivel, así como la cantidad de alumnos máximos en cada curso.
8. El sistema debe permitir al usuario configurador hacer labores de respaldo y recuperación del sistema.
9. El sistema debe permitir al usuario configurador puede crear cuentas directamente.
10. El sistema debe permitir al usuario administrativo formas de cargar información de alumnos, representantes y docentes, en un formato que se establecerá luego.
11. El sistema debe permitir al usuario administrativo las tareas de cargar pagos.
12. El sistema debe permitir al usuario administrativo revisar información de pagos realizados por los representantes.
13. El sistema debe permitir al usuario administrativo verificar los pagos realizados por un representante.
14. El sistema debe permitir al usuario administrativo la emisión de informes que le permitan ver asuntos relacionados con el estado de cuenta de un alumno o grupo de alumnos.
15. El sistema debe permitir al usuario administrativo emitir correos electrónicos, según criterio de selección, a representantes, docentes u otros administrativos.
16. El sistema debe permitir al usuario docente sólo el envío de correos electrónicos a un representante o grupo de representantes.
17. El sistema debe permitir al usuario representante la revisión de su estado de cuenta.
18. El sistema debe permitir al usuario representante la notificación de un pago realizado.

## Casos de Uso:

El diagrama de casos de usos, le indica al lector, cómo entendemos el problema y como el sistema va a cumplir con la lista de características anteriormente enunciada.



Por razones de claridad todos los Casos de Uso “usan” el caso de uso de Guardar/Editar/Recuperar/Eliminar datos de la base de datos.

Relación de los casos de uso con la lista de características: esta relación nos permite ver cuales casos de uso satisfarán la lista de requerimientos

|  |  |
| --- | --- |
| Caso de Uso | Lista de Características |
| 1.- Guardar / editar / recuperar / eliminar de la Base de Datos | * El sistema debe permitir guardar / editar / recuperar información de la base de datos. (1) * El sistema debe contar con un usuario de tipo configurador. (3) * El sistema debe ser capaz de representar a un alumno y sus representantes. (4) |
| 2.- Ingreso y autenticación | * El sistema debe ser capaz de identificar un usuario y discriminar su funcionalidad de acuerdo a su rol. (2) |
| 3.- Auto Registro | * El sistema debe contar con una forma que permita registrarse e ingresar al mismo a un personal administrativo, a un docente y a un representante. (6) |
| 4.- Configura Usuarios | * El sistema debe permitir al usuario configurador puede crear cuentas directamente. (9) |
| 5.- Configura Períodos | * El sistema debe permitir al usuario configurador la carga de información del año lectivo, de la cantidad de niveles, cursos y secciones en cada nivel, así como la cantidad de alumnos máximos en cada curso. (7) |
| 6.- Configura Nivel | * El sistema debe permitir al usuario configurador la carga de información del año lectivo, de la cantidad de niveles, cursos y secciones en cada nivel, así como la cantidad de alumnos máximos en cada curso. (7) |
| 7.- Configura Cursos | * El sistema debe permitir al usuario configurador la carga de información del año lectivo, de la cantidad de niveles, cursos y secciones en cada nivel, así como la cantidad de alumnos máximos en cada curso. (7) |
| 8.- Configura sección | * El sistema debe permitir al usuario configurador la carga de información del año lectivo, de la cantidad de niveles, cursos y secciones en cada nivel, así como la cantidad de alumnos máximos en cada curso. (7) |
| 9.- Configura Pagos | * El sistema debe permitir al usuario administrativo las tareas de cargar pagos. (11) |
| 10.- Verifica Pagos | * El sistema debe permitir al usuario administrativo revisar información de pagos realizados por los representantes. (13) |
| 11.- Configura Profesor | * El sistema debe permitir al usuario administrativo formas de cargar información de alumnos, representantes y docentes, en un formato que se establecerá luego. (10) |
| 12.- Configura Alumno | * El sistema debe permitir al usuario administrativo formas de cargar información de alumnos, representantes y docentes, en un formato que se establecerá luego. (10) * El sistema debe ser capaz de asociar a un alumno a un nivel/curso/sección. (5) |
| 13.- Emite Informes | * El sistema debe permitir al usuario administrativo revisar información de pagos realizados por los representantes. (12) * El sistema debe permitir al usuario administrativo la emisión de informes que le permitan ver asuntos relacionados con el estado de cuenta de un alumno o grupo de alumnos. (14) |
| Maneja emisión de correos | * El sistema debe permitir al usuario administrativo emitir correos electrónicos, según criterio de selección, a representantes, docentes u otros administrativos. (15) * El sistema debe permitir al usuario docente sólo el envío de correos electrónicos a un representante o grupo de representantes. (16) |
| Realiza Pagos | * El sistema debe permitir al usuario representante la notificación de un pago realizado. (18) |
| Revisa saldo | * El sistema debe permitir al usuario representante la notificación de un pago realizado. (17) |

**Descripción de los Casos de Uso:**

|  |  |  |
| --- | --- | --- |
| **CU-001** | **El sistema debe permitir guardar / editar / recuperar información de la base de datos** | |
| **Dependencias** |  | |
| **Actores** | Sistema | |
| **Precondición** | Debe existir una base de datos configurada y funcional | |
| **Descripción** | El sistema debe comunicarse con la base de datos para escribir / recuperar / editar / eliminar | |
| **Flujo Normal** | **Paso** | **Acción** |
| 1 | El sistema hace una petición de conexión a la base de datos |
| 2 | Al levantar debe verificar la existencia de un usuario configurador |
| 3 | Debe contar con formas de representar a un alumno, docente, representante y configurador |
| 4 | Ejecuta la función solicitada (leer / escribir / eliminar / actualizar) información en una o más tablas de la Base de Datos |
|  |  |
| **Post condición** | La respuesta de la Base de datos | |
| **Flujo Alterno 1** | **Paso** | **Acción** |
| 1.1 | La Base de Datos no se encuentra operativa |
| 1.2 | Emitir mensaje sobre la situación de desconexión a la base de datos |
| **Flujo Alterno 2** | **Paso** | **Acción** |
| 2.1 | No existe un usuario configurador |
| 2.2 | El sistema crea un usuario configurador por defecto |

|  |  |  |
| --- | --- | --- |
| **CU-002** | **Ingreso y Autenticación** | |
| **Dependencias** | CU-001 | |
| **Actores** | Configurador, Docente, Representante, Administrativo | |
| **Precondición** | Debe existir una base de datos configurada y funcional / El usuario debe estar previamente registrado | |
| **Descripción** | El sistema debe identificar y autenticar a un usuario | |
| **Flujo Normal** | **Paso** | **Acción** |
| 1 | El usuario ingresa datos credenciales de autenticación |
| 2 | El sistema va a la base de datos y verifica los datos |
| 3 | Lo identifica y le da ingreso, de acuerdo a su rol |
|  |  |
|  |  |
| **Post condición** | El usuario puede trabajar según su rol | |
| **Flujo Alterno 1** | **Paso** | **Acción** |
| 1.1 | EL usuario ingresa mal las credenciales |
| 1.2 | Emitir mensaje sobre la negación de ingreso |
| **Flujo Alterno 2** | **Paso** | **Acción** |
| 2.1 | El usuario no está registrado |
| 2.2 | Emitir mensaje indicando que se debe registrar |

|  |  |  |
| --- | --- | --- |
| **CU-003** | **Auto-registro** | |
| **Dependencias** | CU-001 | |
| **Actores** | Administrativo, Docente, Representante | |
| **Precondición** | Debe existir una base de datos configurada y funcional /El configurador debe previamente haber cargado la información básica del usuario | |
| **Descripción** | El sistema debe permitir a un usuario auto registrarse en el sistema | |
| **Flujo Normal** | **Paso** | **Acción** |
| 1 | El usuario ingresa sus datos de ingreso (C.I., Correo Electrónico y Clave que desea usar) |
| 2 | El sistema verifica que la C.I. del usuario existe |
| 3 | Le valida sus datos |
| 4 | Lo ingresa a la base de datos como usuario |
|  | 5 | Le presenta la forma de acuerdo a su rol |
| **Post condición** | Presenta la pantalla de acuerdo a su rol | |
| **Flujo Alterno 2** | **Paso** | **Acción** |
| 2.1 | La C.I ingresada no existe |
| 2.2 | Le informa al usuario la situación indicando que debe contactar al configurador |
| **Flujo Alterno 3** | 3.1 | El correo y/o clave ya existen |
| 3.2 | Le notifica al usuario la condición, para que corrija |

|  |  |  |
| --- | --- | --- |
| **CU-004** | **Configura usuarios** | |
| **Dependencias** | CU-001 | |
| **Actores** | Configurador | |
| **Precondición** | Debe existir una base de datos configurada y funcional. | |
| **Descripción** | El usuario configurador ingresa los datos básicos de un usuario Administrativo, Docente (Nombre, Apellido, C.I., rol) | |
| **Flujo Normal** | **Paso** | **Acción** |
| 1 | El sistema le presenta la forma correspondiente al configurador |
| 2 | El sistema verifica que la C.I. del usuario existe |
| 3 | El configurador ingresa nombre, apellido y rol |
| 4 | El sistema lo almacena en la base de datos |
| **Post condición** | Notificación que el usuario ha sido creado con éxito | |
| **Flujo Alterno 2** | **Paso** | **Acción** |
| 2.1 | La C.I ingresada ya existe |
| 2.2 | Le informa al configurador la situación |
|  | 2.3 | Muestra la información de la C.I. cargada |

|  |  |  |
| --- | --- | --- |
| **CU-005** | **Configura períodos** | |
| **Dependencias** | CU-001 | |
| **Actores** | Configurador | |
| **Precondición** | Debe existir una base de datos configurada y funcional / Debe existir una forma para la administración del período | |
| **Descripción** | El usuario configurador ingresa el nombre del nuevo período escolar | |
| **Flujo Normal** | **Paso** | **Acción** |
| 1 | El sistema le presenta la forma correspondiente al configurador |
| 2 | El sistema verifica que el nombre del período no existe |
| 3 | El sistema lo almacena en la base de datos |
| **Post condición** | Notificación que el período ha sido creado con éxito | |
| **Flujo Alterno 2** | **Paso** | **Acción** |
| 2.1 | El nombre ingresado ya existe |
| 2.2 | Le informa al configurador la situación |
| 2.3 | Muestra la información del período ya cargada |

|  |  |  |
| --- | --- | --- |
| **CU-006** | **Configura Nivel** | |
| **Dependencias** | CU-001 | |
| **Actores** | Configurador | |
| **Precondición** | Debe existir una base de datos configurada y funcional. / Debe existir una forma para la administración del nivel | |
| **Descripción** | El usuario configurador ingresa los datos de configuración de un nivel. | |
| **Flujo Normal** | **Paso** | **Acción** |
| 1 | El sistema le presenta la forma correspondiente al configurador |
| 2 | El configurador ingresa nombre del nivel |
| 3 | El sistema verifica que el nivel ingresado no existe |
| 4 | El sistema lo almacena en la base de datos |
| **Post condición** | Notificación que el nivel ha sido creado con éxito | |
| **Flujo Alterno 2** | **Paso** | **Acción** |
| 2.1 | El nivel ingresado ya existe |
| 2.2 | Le informa al configurador la situación |
| 2.3 | Muestra la información del nivel ya cargada |

|  |  |  |
| --- | --- | --- |
| **CU-007** | **Configura Curso** | |
| **Dependencias** | CU-001 | |
| **Actores** | Configurador | |
| **Precondición** | Debe existir una base de datos configurada y funcional. / Debe existir una forma para la administración del Curso | |
| **Descripción** | El usuario configurador ingresa los datos de configuración del curso. | |
| **Flujo Normal** | **Paso** | **Acción** |
| 1 | El sistema le presenta la forma correspondiente al configurador. |
| 2 | El configurador ingresa nombre del curso. |
| 3 | El sistema verifica que el presente curso no existe. |
| 4 | El sistema lo almacena en la base de datos |
| **Post condición** | Notificación que el curso ha sido creado con éxito | |
| **Flujo Alterno 2** | **Paso** | **Acción** |
| 2.1 | EL curso ingresado ya existe |
| 2.2 | Le informa al configurador la situación |
| 2.3 | Muestra la información del curso ya cargada |

|  |  |  |
| --- | --- | --- |
| **CU-008** | **Configura Sección** | |
| **Dependencias** | CU-001 | |
| **Actores** | Configurador | |
| **Precondición** | Debe existir una base de datos configurada y funcional. / Debe existir una forma para la administración de la sección | |
| **Descripción** | El usuario configurador ingresa los datos de configuración de la sección. | |
| **Flujo Normal** | **Paso** | **Acción** |
| 1 | El sistema le presenta la forma correspondiente al configurador. |
| 2 | El configurador ingresa nombre de la sección. |
| 3 | El configurador ingresa la cantidad máxima de alumnos de la sección. |
| 4 | El sistema verifica que la presente Sección no existe. |
| 5 | El sistema la almacena en la base de datos |
| **Post condición** | Notificación que la sección ha sido creado con éxito | |
| **Flujo Alterno 2** | **Paso** | **Acción** |
| 2.1 | La sección ingresada ya existe |
| 2.2 | Le informa al configurador la situación |
| 2.3 | Muestra la información de la sección ya cargada |

|  |  |  |
| --- | --- | --- |
| **CU-009** | **Configura Pagos** | |
| **Dependencias** | CU-001 | |
| **Actores** | Administrativo | |
| **Precondición** | Debe existir una base de datos configurada y funcional. / Debe existir una forma para la administración de los pagos | |
| **Descripción** | El usuario administrativo ingresa los datos de configuración de los pagos. | |
| **Flujo Normal** | **Paso** | **Acción** |
| 1 | El sistema le presenta la forma correspondiente al administrativo. |
| 2 | El administrativo ingresa nombre del pago y monto. |
| 3 | El administrativo asigna el pago a un nivel, curso o sección. |
| 4 | El sistema verifica que el pago no existe. |
| 5 | El sistema lo almacena en la base de datos |
| **Post condición** | Notificación que la sección ha sido creado con éxito | |
| **Flujo Alterno 4** | **Paso** | **Acción** |
| 4.1 | El pago ingresado ya existe |
| 4.2 | Le informa al administrativo la situación |
| 4.3 | Muestra la información del pago ya cargado |

|  |  |  |
| --- | --- | --- |
| **CU-010** | **Verifica Pagos** | |
| **Dependencias** | CU-001 | |
| **Actores** | Administrativo | |
| **Precondición** | Debe existir una base de datos configurada y funcional. / Debe existir una forma para la administración de los pagos | |
| **Descripción** | El usuario administrativo ingresa los datos del pago recibido. | |
| **Flujo Normal** | **Paso** | **Acción** |
| 1 | El sistema le presenta la forma correspondiente al administrativo. |
| 2 | El administrativo revisa la cuenta de banco del colegio para validar que se recibió el pago. |
| 3 | El administrativo coloca el pago como recibido en el sistema |
| 4 | El sistema lo almacena en la base de datos |
| **Post condición** | Notificación que la sección ha sido creado con éxito | |
| **Flujo Alterno** | **Paso** | **Acción** |
| 2.1 | El pago no se ha recibido |
| 2.2 | Deja pendiente el pago |
| 2.3 | El sistema almacena el pago como pendiente |

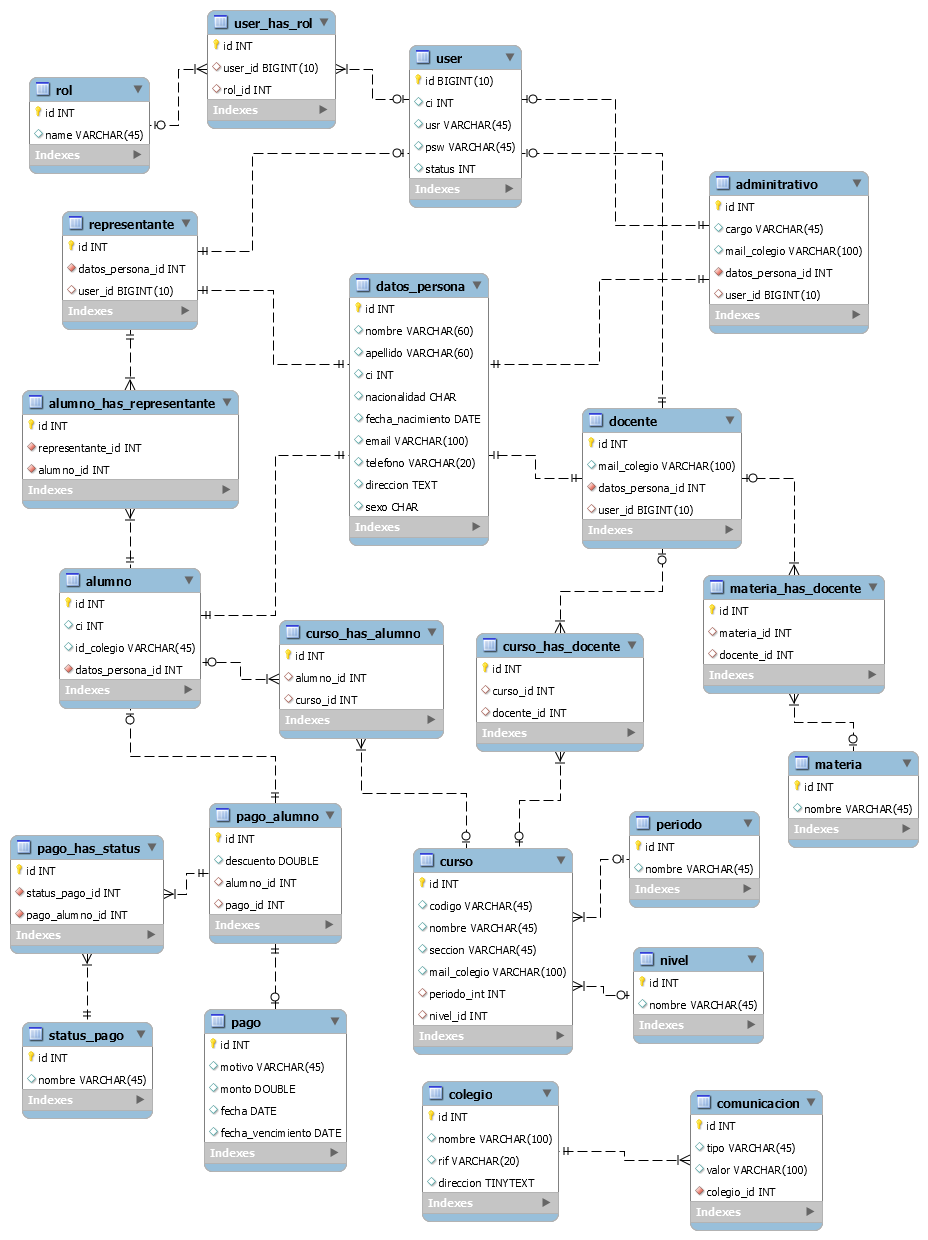
|  |  |  |
| --- | --- | --- |
| **CU-011** | **Configura Docente** | |
| **Dependencias** | CU-001 | |
| **Actores** | Administrativo | |
| **Precondición** | Debe existir una base de datos configurada y funcional. / Debe existir una forma para la administración de los docentes | |
| **Descripción** | El usuario administrativo configura un docente. | |
| **Flujo Normal** | **Paso** | **Acción** |
| 1 | El sistema le presenta la forma correspondiente al administrativo. |
| 2 | El administrativo carga los datos del docente, tanto personales como académicos. |
| 3 | El sistema lo almacena en la base de datos |
| **Post condición** | Notificación que la sección ha sido creado con éxito | |
| **Flujo Alterno** | **Paso** | **Acción** |
| 2.1 | El docente ya existe |
| 2.2 | El sistema informa al administrativo de la situación |
| 2.3 | El sistema muestra la información del docente ya cargado |

## Iteraciones

Caso de Uso N° 1: El sistema debe permitir guardar / editar / recuperar información de la base de datos.

1. Debe existir una base de datos
2. El sistema hace una conexión a esa BD
3. La BD responde conexión realizada
4. SE ejecutan comando de escritura, lectura, actualización y eliminación sobre los datos en la BD

Diseño de la base de datos:



Tablas:

Los campos son todos optativos, a menos que se indique que son obligados

Colegio: datos básicos del colegio:

* + Nombre
  + Rif
  + Dirección

Comunicacion: forma de comunicación con el colegio:

* + Tipo: el tipo de comunicación: teléfono, email, Facebook, twitter, etc
  + Valor: el valor que toma el tipo de comunicación
  + Posee una relación muchos-a-uno con **colegio** a través de la FK: colegio\_id

User: el usuario autorizado a ingresar al sistema:

* + Ci: la cédula de identidad del usuario
  + Usr: el usuario usado para ingresar (será un email del usuario)
  + Psw: el password elegido por el usuario
  + Status: Habilitado (1), Inhabilitado (0)

Rol: indica el (los) roles que puede tener un usuario.

* + Name: el nombre del rol (Administrativo, Representante, Docente, Configurador)

User\_has\_rol: tabla intermedia usada para relacionar el usuario con su(s) rol(es), de esta manera un usuario puede tener más de un rol.

Datos\_persona: los datos personales correspondiente a los diferentes roles del sistema. Esta tabla es de complemento con los diferentes actores del sistema, no puede estar sola:

* + Id: entero, auto-incremental
  + Nombre
  + Apellido
  + Ci: cédula de identidad (en caso de poseer)
  + Nacionalidad: toma un solo carácter (V: venezolano, E: extranjero)
  + Fecha\_nacimiento: usado cuando aplique, para cálculos de edad
  + Email: en caso de poseer
  + Teléfono: en caso de poseer
  + Dirección: en caso de ser notificada
  + Sexo: para lo relacionado con estadísticas (valores: M o F)

Administrativo: tabla para identificar un actor de tipo administrativo:

* + Cargo: el cargo ejercido
  + Mail\_colegio: email asignado por el colegio
  + Posee una relación uno-a-uno con **datos\_persona**, a través de la FK: datos\_persona\_id (obligatoria)
  + Posee una relación uno-a-uno con **user**, a través de la FK: user\_id

Docente: tabla para identificar un actor de tipo docente:

* + Mail\_colegio: email asignado por el colegio
  + Posee una relación uno-a-uno con **datos\_persona**, a través de la FK: datos\_persona\_id (obligatoria)
  + Posee una relación uno-a-uno con **user**, a través de la FK: user\_id

Representante: tabla para identificar un actor de tipo representante:

* + Posee una relación uno-a-uno con **datos\_persona**, a través de la FK: datos\_persona\_id (obligatoria)
  + Posee una relación uno-a-uno con **user**, a través de la FK: user\_id

Alumno: tabla para identificar a un alumno, no es actor del sistema, ya que no posee usuario de ingreso:

* + Id: campo autoincremental, para asegurar un id interno de la base de datos
  + Ci: cédula de identidad, en caso de poseer, es un campo UNIQUE para evitar repeticiones
  + Id\_colegio: identificación que asigna el colegio al alumno, es un campo UNIQUE para evitar repeticiones
  + Posee una relación uno-a-uno con **datos\_persona**, a través de la FK: datos\_persona\_id (obligatoria)

Alumno\_has\_representante: tabla para relacionar los alumnos con sus representantes. Se puede dar el caso que un representante lo sea de más de un alumno, así mismo un alumno puede estar representado por más de un reprsentante.

Materia: tabla usada para identificar las materias (asignaturas) impartidas en el colegio

* + Nombre: el nombre de la materia

Materia\_has\_docente: tabla que relaciona una (o más) materia(s) con un (o más) docente(s). Esto se hace ya que se puede dar el caso que un docente tenga más de una materia, o una materia sea impartida por más de un docente.

Periodo: tabla que representa los diferentes períodos académicos.

* + Nombre: el nombre asignado al período, este nombre es clave única para evitar repeticiones de nombre, se almacenará llevando todo a mayúsculas.

Nivel: tabla que representa los diferentes niveles educativos del colegio: maternal, pre-escolar, primaria, bachillerato.

* + Nombre: el nombre asignado al nivel (UNIQUE, para evitar repeticiones)

Curso: tabla que representa un curso del colegio: grado-sección

* + Codigo: usado en caso de que el colegio asigne códigos a los cursos
  + Nombre: nombre del curso (primer grado, tercer año, etc)
  + Seccion: usado para cuando un curso esté dividido en secciones
  + Mail\_colegio: usado en caso que se asigne un mail a un curso
  + Posee una relación muchos-a-uno con **periodo** a través de la FK: periodo\_id
  + Posee una relación muchos-a-uno con **nivel** a través de la FK: nivel\_id

Curso\_has\_docente: tabla para relacionar los docentes con el curso, se puede dar el caso que un docente esté en más de curso, además que un curso puede tener más de un docente.

Curso\_has\_alumno: tabla para relacionar los alumnos con el curso, cada alumno está en un curso, pero al cambiar el período, ese alumno aparecerá en el (los) curso(s) anterior(es) también. Permite llevar un histórico del alumno.

Pago: el pago a realizar de un concepto insertado al sistema:

* + Motivo: nombre del pago a realizar: inscripción, mensualidad, etc
  + Monto: cantidad de Bs. del pago a realizar
  + Fecha: fecha en que el pago se registró en el sistema
  + Fecha\_vencimiento: fecha en que vence el pago

Pago\_alumno: el pago a realizar por un representante:

* + Descuento: en caso que el pago realizado tenga un descuento, se refleja aquí.
  + Posee una relación uno-a-uno con **alumno** a través de la FK: alumno\_id
  + Posee una relación uno-a-uno con **pago** a través de la FK: pago\_id

La relación indica que sólo se puede asignar al alumno un determinado pago, es decir no se le puede imputar un mismo pago más de una vez.

Status\_pago: tabla para almacenar los diferentes status que puede tener un pago

* + Nombre: el nombre del status de pago
  + valor: el estado en que se encuentra el pago: Pendiente (0), Pagado en tiempo (1), pagado en atraso (2), Exonerado (3), con descuento (4)

Pago\_has\_status: tabla para identificar los diferentes status que puede tener un pago, se hace así ya que un pago puede tener más de un status, ejemplo: con descuento y pendiente

* + Posee una relación muchos-a-uno con **pago\_alumno** a través de la FK: pago\_alumno\_id
  + Posee una relación muchos-a-uno con **status\_pago** a través de la FK: status\_pago\_id