

Contenido

6. Modelo de datos.....	1
6.1. Diagramas de tablas	1
6.1.1. Explicación del diagrama	2
6.1.2. Script BD	4
6.2. Otros elementos de almacenamiento	20
6.2.1. Diseño y explicación elemento 1	20
6.2.2. Diseño y explicación elemento 2	21
6.2.3. Diseño y explicación elemento 3	22
6.2.4. Diseño y explicación elemento 4	22
6.2.5. Diseño y explicación elemento 5	23
6.2.6. Diseño y explicación elemento 6	23
6.2.7. Diseño y explicación elemento 7	24

6. Modelo de datos

6.1. Diagramas de tablas

<Copia de la imagen del diagrama de tablas realizado en Toad. El diagrama se generará automáticamente a partir del diagrama E/R. Se pueden ver reglas de conversión en: Help | PER-LER Conversion Information. Se tomar decisiones no automáticas sobre: relaciones de herencia y relaciones 1:1. Para decidir la conversión de las relaciones de herencia, sobre el diagrama E/R, doble clic sobre la relación e ir a la pestaña Generation. Para decidir la conversión de las relaciones 1:1 Toad tiene en cuenta sobre qué entidad se empezó a dibujar la línea de la relación, por lo que si la conversión no es la correcta cambiar el origen de la relación en el diagrama E/R y volver a realizar la conversión. Para generar el modelo físico (diagrama de tablas): se abre el modelo lógico (diagrama E/R), Model | Convert Model | Simple Conversion y seleccionar una base de datos (no Universal). Una vez realizada la conversión automática se añaden como campos de las tablas que provienen de una relación n:m, los atributos de la relación. Finalmente es posible desnormalizar por necesidades de diseño: rendimiento, seguridad,...>

En esta relación se ha decidido hacer que el modelo físico coincida con el modelo lógico en primer lugar porque facilita la legibilidad, ya que, si no se elige esta opción se crean cuatro tablas de la relación Preso realiza Actividad, una por cada tipo. Además, esta relación es n:m por lo que es más sencillo añadir sus atributos en una sola tabla que en cuatro.

Hereda1

General
Generation
Descendents
To Do
Description

Physical Implementation

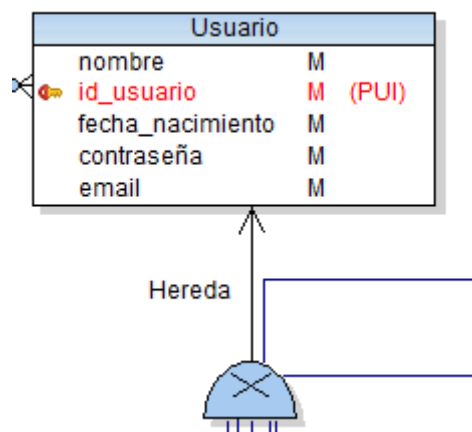
☐ Single Table - Parent Inherits All Children

Discriminator

☐ N - 1 Tables - Each Child Inherits the Parent

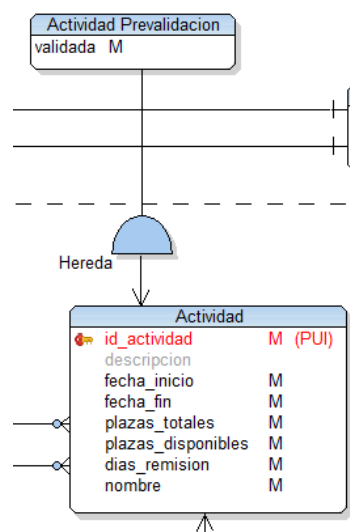
☒ N Tables - Physical Model Matches Logical

- **Relación de herencia Asistente Social, Director, Empleado, Preso, Profesor, Responsable Actividad 3 y Responsable Taller heredan de Usuario:**



En esta relación se ha decidido hacer, como en la anterior, que el modelo físico coincida con el lógico para que aparezca la entidad Usuario y de esta manera facilitar la legibilidad y comprensión del sistema, ya que, se puede ver cada usuario con su contraseña y esto facilitaría la programación del acceso de usuarios al sistema.

- **Relación de herencia Actividad Prevalidación hereda de Actividad:**



La entidad Actividad Prevalidación es una entidad encargada de almacenar las modificaciones de las actividades antes de ser validadas. Por tanto, se ha decidido que coincida con el lógico para facilitar el funcionamiento de la aplicación.

Se ha discutido si es necesaria la desnormalización para mejorar el rendimiento. Si una tabla fuese demasiado grande, cada tupla ocuparía varios bloques de memoria y si sobre ella detectásemos atributos sobre los que no se accederá con frecuencia o que no son necesarios para la actividad base del sistema podría ser buena idea desnormalizar creando dos tablas. Una principal con los datos a los que se accede con frecuencia cuya misión será no entorpecer otras consultas para las que el tamaño de sus tuplas podría estar haciendo de cuellos de botella, y otra con el resto de los atributos unidas ambas por una relación 1:1.

Sin embargo, hemos llegado a la conclusión de que, en nuestro caso, no es rentable desnormalizar, ya que, aunque la duplicación en varias tablas de columnas solicitadas con frecuencia sería menos costosa, el número de entidades es pequeño, por lo que el tiempo de acceso de las consultas a esas columnas a través de uniones no requiere un gran rendimiento, y además tampoco existe una entidad con numerosos atributos. Así además se evita la redundancia de datos.

6.1.2. Script BD

<Copia del contenido del fichero obtenido tras la generación automática en Toad del script de BD. Se utiliza la opción Model / Generate DDL script / Run.>

/*

Created: 16/01/2022

Modified: 16/01/2022

Project: ISA_2

Model: Physical Model

Database: MS SQL Server 2019

*/

-- Create tables section -----

-- Table Preso

CREATE TABLE [Preso]

(

[duracion_condena] Int NOT NULL,

[expediente] Char(256) NOT NULL,

[id_usuario] Int NOT NULL,

```
[id_usuario] Int NULL
```

```
)
```

```
go
```

```
-- Create indexes for table Preso
```

```
CREATE INDEX [IX_Ayuda] ON [Preso] ([id_usuario])
```

```
go
```

```
-- Add keys for table Preso
```

```
ALTER TABLE [Preso] ADD CONSTRAINT [Unique_Identifier1] PRIMARY KEY ([id_usuario])
```

```
go
```

```
-- Table Actividad
```

```
CREATE TABLE [Actividad]
```

```
(
```

```
[id_actividad] Int NOT NULL,
```

```
[descripcion] Char(256) NULL,
```

```
[fecha_inicio] Date NOT NULL,
```

```
[fecha_fin] Date NOT NULL,
```

```
[plazas_totales] Bigint NOT NULL,
```

```
[plazas_disponibles] Bigint NOT NULL,
```

```
[dias_remision] Int NOT NULL,
```

```
[nombre] Char(25) NOT NULL
```

```
)
```

```
go
```

```
-- Add keys for table Actividad
```

```
ALTER TABLE [Actividad] ADD CONSTRAINT [Unique_Identifier4] PRIMARY KEY ([id_actividad])
```

```
go
```

```
-- Table Curso FP
```

```
CREATE TABLE [Curso FP]
```

```
(
```

```
[dias_remision] Int NOT NULL,
```

```
[id_actividad] Int NOT NULL
```

```
)
```

```
go
```

```
-- Add keys for table Curso FP
```

```
ALTER TABLE [Curso FP] ADD CONSTRAINT [Unique_Identifier5] PRIMARY KEY ([id_actividad])
```

```
go
```

```
-- Table Taller
```

```
CREATE TABLE [Taller]
```

```
(
```

```
[minimo_horas] Int NOT NULL,
```

```
[dias_remision] Int NOT NULL,
```

```
[codigo_puerta] Int NOT NULL,
```

```
[id_actividad] Int NOT NULL
```

```
)
```

```
go
```

```
-- Add keys for table Taller
```

```
ALTER TABLE [Taller] ADD CONSTRAINT [Unique_Identifier6] PRIMARY KEY ([id_actividad])
```

```
go
```

```
-- Table Trabajo
```

```
CREATE TABLE [Trabajo]
```

```
(  
[horas_remision] Int NOT NULL,  
[codigo_puerta] Int NOT NULL,  
[id_actividad] Int NOT NULL  
)
```

```
go
```

```
-- Add keys for table Trabajo
```

```
ALTER TABLE [Trabajo] ADD CONSTRAINT [Unique_Identifier7] PRIMARY KEY ([id_actividad])
```

```
go
```

```
-- Table Asistente Social
```

```
CREATE TABLE [Asistente Social]
```

```
(  
[id_usuario] Int NOT NULL  
)
```

```
go
```

```
-- Add keys for table Asistente Social
```

```
ALTER TABLE [Asistente Social] ADD CONSTRAINT [Unique_Identifier9] PRIMARY KEY ([id_usuario])
```

```
go
```

```
-- Table Director
```

```
CREATE TABLE [Director]
```

```
(  
[id_usuario] Int NOT NULL  
)  
go
```

```
-- Add keys for table Director
```

```
ALTER TABLE [Director] ADD CONSTRAINT [Unique_Identifier10] PRIMARY KEY ([id_usuario])  
go
```

```
-- Table Centro Educación a Distancia
```

```
CREATE TABLE [Centro Educación a Distancia]  
(  
[nombre] Char(256) NOT NULL,  
[id_centro] Int NOT NULL,  
[contraseña] Char(256) NOT NULL  
)  
go
```

```
-- Add keys for table Centro Educación a Distancia
```

```
ALTER TABLE [Centro Educación a Distancia] ADD CONSTRAINT [Unique_Identifier13] PRIMARY KEY  
([id_centro])  
go
```

```
-- Table Usuario
```

```
CREATE TABLE [Usuario]
```



```
(  
[nombre] Char(50) NOT NULL,  
[id_usuario] Int NOT NULL,  
[fecha_nacimiento] Date NOT NULL,  
[contraseña] Char(256) NOT NULL,  
[email] Char(256) NOT NULL  
)  
go
```

```
-- Add keys for table Usuario
```

```
ALTER TABLE [Usuario] ADD CONSTRAINT [Unique_Identifier23] PRIMARY KEY ([id_usuario])  
go
```

```
-- Table Departamento
```

```
CREATE TABLE [Departamento]  
(  
[id_departamento] Int NOT NULL,  
[nombre_departamento] Char(50) NOT NULL,  
[descripcion] Char(256) NULL,  
[id_usuario] Int NULL  
)  
go
```

```
-- Create indexes for table Departamento
```

```
CREATE INDEX [IX_Coordina] ON [Departamento] ([id_usuario])  
go
```

```
-- Add keys for table Departamento
```

```
ALTER TABLE [Departamento] ADD CONSTRAINT [Unique_Identifier24] PRIMARY KEY ([id_departamento])  
  
go
```

```
-- Table Profesor
```

```
CREATE TABLE [Profesor]  
  
(  
  
[id_usuario] Int NOT NULL  
  
)  
  
go
```

```
-- Add keys for table Profesor
```

```
ALTER TABLE [Profesor] ADD CONSTRAINT [Unique_Identifier27] PRIMARY KEY ([id_usuario])  
  
go
```

```
-- Table Empleado
```

```
CREATE TABLE [Empleado]  
  
(  
  
[id_departamento] Int NOT NULL,  
[id_usuario] Int NOT NULL  
  
)  
  
go
```

```
-- Add keys for table Empleado
```

```
ALTER TABLE [Empleado] ADD CONSTRAINT [Unique_Identifier28] PRIMARY KEY  
([id_departamento],[id_usuario])  
  
go
```

-- Table Asignatura

```
CREATE TABLE [Asignatura]
(
[id_asignatura] Int NOT NULL,
[nombre] Char(256) NOT NULL,
[descripcion] Char(256) NULL,
[dias_remision] Int NOT NULL,
[id_curso] Int NOT NULL,
[id_actividad] Int NOT NULL
)
go
```

-- Add keys for table Asignatura

```
ALTER TABLE [Asignatura] ADD CONSTRAINT [Unique_Identifier29] PRIMARY KEY
([id_asignatura],[id_curso],[id_actividad])
go
```

-- Table Curso a Distancia

```
CREATE TABLE [Curso a Distancia]
(
[nombre] Char(256) NOT NULL,
[id_curso] Int NOT NULL,
[descripcion] Char(256) NULL,
[id_centro] Int NOT NULL,
[id_actividad] Int NOT NULL
)
go
```

-- Create indexes for table Curso a Distancia

```
CREATE INDEX [IX_Se compone] ON [Curso a Distancia] ([id_centro])
```

go

-- Add keys for table Curso a Distancia

```
ALTER TABLE [Curso a Distancia] ADD CONSTRAINT [Unique_Identifier30] PRIMARY KEY  
([id_curso],[id_actividad])
```

go

-- Table Registro E/S

```
CREATE TABLE [Registro E/S]
```

(

[id_registro] Bigint NOT NULL,

[hora_entrada] Datetime NOT NULL,

[hora_salida] Datetime NOT NULL,

[fecha] Date NOT NULL,

[validado] Bit NOT NULL,

[lugar_fichaje] Char(256) NULL,

[id_usuario] Int NOT NULL,

[id_usuario] Int NOT NULL,

[id_usuario] Int NOT NULL

)

go

-- Add keys for table Registro E/S

```
ALTER TABLE [Registro E/S] ADD CONSTRAINT [Unique_Identifier31] PRIMARY KEY  
([id_registro],[id_usuario],[id_usuario],[id_usuario])
```

go

-- Table Actividad Prevalidacion

CREATE TABLE [Actividad Prevalidacion]

(

[validada] Bit NOT NULL,

[id_actividad] Int NOT NULL

)

go

-- Add keys for table Actividad Prevalidacion

ALTER TABLE [Actividad Prevalidacion] ADD CONSTRAINT [Unique_Identifier32] PRIMARY KEY ([id_actividad])

go

-- Table Clase

CREATE TABLE [Clase]

(

[horario] Datetime NOT NULL,

[id_actividad] Int NOT NULL

)

go

-- Add keys for table Clase

ALTER TABLE [Clase] ADD CONSTRAINT [Unique_Identifier33] PRIMARY KEY ([id_actividad])

go

-- Table Tipo de Condena

```
CREATE TABLE [Tipo de Condena]
```

```
(  
[descripcion] Char(256) NULL,  
[fecha] Date NOT NULL,  
[nombre] Char(256) NOT NULL  
)
```

```
go
```

```
-- Add keys for table Tipo de Condena
```

```
ALTER TABLE [Tipo de Condena] ADD CONSTRAINT [Unique_Identifier34] PRIMARY KEY ([nombre])
```

```
go
```

```
-- Table Visita
```

```
CREATE TABLE [Visita]
```

```
(  
[pdf] Char(256) NOT NULL,  
[id_usuario] Int NOT NULL,  
[id_usuario] Int NOT NULL  
)
```

```
go
```

```
-- Add keys for table Visita
```

```
ALTER TABLE [Visita] ADD CONSTRAINT [Unique_Identifier35] PRIMARY KEY ([id_usuario],[id_usuario])
```

```
go
```

```
-- Table Informe
```

```
CREATE TABLE [Informe]
(
[id_departamento] Int NOT NULL,
[id_usuario] Int NOT NULL
)
go
```

```
-- Add keys for table Informe
```

```
ALTER TABLE [Informe] ADD CONSTRAINT [Unique_Identifier36] PRIMARY KEY
([id_departamento],[id_usuario])
go
```

```
-- Table Preso_Actividad
```

```
CREATE TABLE [Preso_Actividad]
(
[id_actividad] Int NOT NULL,
[id_usuario] Int NOT NULL,
[horas_trabajadas] Int NOT NULL,
[es_realizada] Bit NOT NULL
)
go
```

```
-- Table Taller_Profesor
```

```
CREATE TABLE [Taller_Profesor]
(
[id_actividad] Int NOT NULL,
[id_usuario] Int NOT NULL
)
```

go

-- Table Preso_Actividad

CREATE TABLE [Preso_Actividad]

(
[id_actividad] Int NOT NULL,
[id_usuario] Int NOT NULL,
[prioridad] Int NOT NULL
)

go

-- Table Profesor_Clase

CREATE TABLE [Profesor_Clase]

(
[id_actividad] Int NOT NULL,
[id_usuario] Int NOT NULL
)

go

-- Table Preso_Tipo de Condena

CREATE TABLE [Preso_Tipo de Condena]

(
[nombre] Char(256) NOT NULL,
[id_usuario] Int NOT NULL
)

go

-- Table Empleado_Taller


```
CREATE TABLE [Empleado_Taller]
(
[id_departamento] Int NOT NULL,
[id_actividad] Int NOT NULL,
[id_usuario] Int NOT NULL
)
go
```

-- Table Empleado_Trabajo

```
CREATE TABLE [Empleado_Trabajo]
(
[id_departamento] Int NOT NULL,
[id_actividad] Int NOT NULL,
[id_usuario] Int NOT NULL
)
go
```

-- Table Registro E/S_Informe

```
CREATE TABLE [Registro E/S_Informe]
(
[id_registro] Bigint NOT NULL,
[id_departamento] Int NOT NULL,
[id_usuario] Int NOT NULL,
[id_usuario] Int NOT NULL,
[id_usuario] Int NOT NULL,
[id_usuario] Int NOT NULL
)
go
```

-- Table Empleado_Registro E/S

CREATE TABLE [Empleado_Registro E/S]

(
[id_departamento] Int NOT NULL,
[id_registro] Bigint NOT NULL,
[id_usuario] Int NOT NULL,
[id_usuario] Int NOT NULL,
[id_usuario] Int NOT NULL,
[id_usuario] Int NOT NULL
)

go

-- Create foreign keys (relationships) section -----

ALTER TABLE [Preso] ADD CONSTRAINT [Ayuda] FOREIGN KEY ([id_usuario]) REFERENCES [Asistente Social]
([id_usuario]) ON UPDATE NO ACTION ON DELETE NO ACTION

go

ALTER TABLE [Empleado] ADD CONSTRAINT [Trabaja] FOREIGN KEY ([id_departamento]) REFERENCES
[Departamento] ([id_departamento]) ON UPDATE NO ACTION ON DELETE NO ACTION

go

ALTER TABLE [Curso a Distancia] ADD CONSTRAINT [Se compone] FOREIGN KEY ([id_centro]) REFERENCES
[Centro Educación a Distancia] ([id_centro]) ON UPDATE NO ACTION ON DELETE NO ACTION

go

```
ALTER TABLE [Asignatura] ADD CONSTRAINT [Contiene] FOREIGN KEY ([id_curso], [id_actividad])  
REFERENCES [Curso a Distancia] ([id_curso], [id_actividad]) ON UPDATE NO ACTION ON DELETE NO ACTION  
  
go
```

```
ALTER TABLE [Departamento] ADD CONSTRAINT [Coordina] FOREIGN KEY ([id_usuario]) REFERENCES  
[Director] ([id_usuario]) ON UPDATE NO ACTION ON DELETE NO ACTION  
  
go
```

```
ALTER TABLE [Clase] ADD CONSTRAINT [Tiene] FOREIGN KEY ([id_actividad]) REFERENCES [Curso FP]  
([id_actividad]) ON UPDATE NO ACTION ON DELETE NO ACTION  
  
go
```

```
ALTER TABLE [Registro E/S] ADD CONSTRAINT [Genera] FOREIGN KEY ([id_usuario]) REFERENCES [Profesor]  
([id_usuario]) ON UPDATE NO ACTION ON DELETE NO ACTION  
  
go
```

```
ALTER TABLE [Visita] ADD CONSTRAINT [Genera] FOREIGN KEY ([id_usuario]) REFERENCES [Asistente Social]  
([id_usuario]) ON UPDATE NO ACTION ON DELETE NO ACTION  
  
go
```

```
ALTER TABLE [Visita] ADD CONSTRAINT [Es visitado] FOREIGN KEY ([id_usuario]) REFERENCES [Preso] ([id_usuario]) ON UPDATE NO ACTION ON DELETE NO ACTION
```

go

```
ALTER TABLE [Registro E/S] ADD CONSTRAINT [Genera] FOREIGN KEY ([id_usuario], [id_usuario]) REFERENCES [Visita] ([id_usuario], [id_usuario]) ON UPDATE NO ACTION ON DELETE NO ACTION
```

go

```
ALTER TABLE [Informe] ADD CONSTRAINT [RevisaInformeAsistente] FOREIGN KEY ([id_usuario]) REFERENCES [Director] ([id_usuario]) ON UPDATE NO ACTION ON DELETE NO ACTION
```

go

```
ALTER TABLE [Informe] ADD CONSTRAINT [RevisaInformeProfesores] FOREIGN KEY ([id_departamento]) REFERENCES [Departamento] ([id_departamento]) ON UPDATE NO ACTION ON DELETE NO ACTION
```

go

6.2. Otros elementos de almacenamiento

<Otros elementos archivados necesarios para la aplicación, por ejemplo ficheros de intercambio de información con otros sistemas o plantillas de correo electrónico.>

6.2.1. Diseño y explicación elemento 1

Plantilla correo lista de los presos con 100 o menos días de condena

El siguiente correo se utiliza cuando el último día de mes el sistema ya ha realizado el cálculo de los días de condena de los presos. Tras realizar dicho cálculo el sistema envía el PDF mediante un correo electrónico al director para informarle de los presos que tienen 100 o menos días de condena. El formato será el siguiente:

Lista de presos con 100 o menos días de condena:

Preso 1:

Nombre: <nombre>
Número de preso: <id_usuario>
Tipo de condena. <tipo_condena>
Días restantes: <duración_condena>

Preso 2:

Nombre: <nombre>
Número de preso: <id_usuario>
Tipo de condena. <tipo_condena>
Días restantes: <duración_condena>

.
.
.

Preso n:

Nombre: <nombre>
Número de preso: <id_usuario>
Tipo de condena. <tipo_condena>
Días restantes: <duración_condena>

6.2.2. Diseño y explicación elemento 2

Plantilla informe visitas del Asistente

La siguiente plantilla será la que rellenarán los asistentes sociales cada vez que realicen la visita de un preso en la cárcel y será un documento PDF. El formato será el siguiente:

Informe de visitas a presos por los asistentes:

Asistente 1: <nombre>

Preso 1:

Nombre: <nombre>
Número de preso: <id_usuario>
Tipo de condena. <tipo_condena>
Días restantes: <duración_condena>
Actividad que está realizando: <actividades_asignadas>
Actividades que ha realizado: <actividades_realizadas>

Preso 2:

Nombre: <nombre>
Número de preso: <id_usuario>
Tipo de condena. <tipo_condena>
Días restantes: <duración_condena>
Actividad que está realizando: <actividades_asignadas>
Actividades que ha realizado: <actividades_realizadas>

.
.

.

.

Preso n:

Nombre: <nombre>

Número de preso: <id_usuario>

Tipo de condena. <tipo_condena>

Días restantes: <duración_condena>

Actividad que está realizando: <actividades_asignadas>

Actividades que ha realizado: <actividades_realizadas>

Asistente 2: <nombre>

Preso 1:

.

.

.

.

.

.

.

Asistente n: <nombre>

Preso 1:

.

.

.

.

6.2.3. Diseño y explicación elemento 3

Plantilla de correo electrónico para informar de la actualización de nuevas actividades

Este correo electrónico se utiliza cuando el jefe de coordinación de actividades ha validado las actividades que han dado de alta, han modificado o eliminado los empleados del departamento de coordinación de actividades.

Estimados reclusos,

Mediante este correo se les informa de que las actividades <nombre>, <nombre>, ..., <nombre> han sufrido los siguientes cambios [en este fragmento se incluirán las modificaciones de las actividades si los hubiese], las siguientes actividades: <nombre>, <nombre>, ..., <nombre> han sido añadidas para que os podáis inscribir y las siguientes actividades: <nombre>, <nombre>, ..., <nombre> han sido eliminadas y ya no es posible inscribirse para realizarlas.

Muchas gracias, DPTO. Coordinación de Actividades

6.2.4. Diseño y explicación elemento 4

Plantilla del informe de acceso de los Asistentes a la prisión

Esta plantilla será la utilizada para generar el informe que posteriormente se enviará por correo electrónico al Director de la prisión para que tenga la información y valide las entradas y salidas de los Asistentes a la prisión. La plantilla será la siguiente:

Informe del acceso de los Asistentes a la prisión

El asistente <nombre> con id <id_usuario> ha ingresado en el recinto a las <hora_entrada> y ha salido a las <hora_salida>.

El asistente <nombre> con id <id_usuario> ha ingresado en el recinto a las <hora_entrada> y ha salido a las <hora_salida>.

.

.

.

El asistente <nombre> con id <id_usuario> ha ingresado en el recinto a las <hora_entrada> y ha salido a las <hora_salida>.

6.2.5. Diseño y explicación elemento 5

Plantilla de correo electrónico que sugiere presos a los Asistente

El siguiente correo electrónico será enviado a los asistentes por el sistema para informarles de los presos que llevan más de seis semanas sin recibir una visita y sugerirles una.

Estimados asistentes,

Los presos con los siguientes id <id_usuario>, <id_usuario>, ..., <id_usuario> llevan más de seis semanas sin recibir ninguna visita. El propósito de este mensaje es sugerirles que les asignen visitas en los próximos días.

Muchas gracias, un saludo.

6.2.6. Diseño y explicación elemento 6

Plantilla de la asignación de las actividades

Esta plantilla será utilizada para generar un PDF cuando el sistema realice las asignaciones de las actividades los jueves por la mañana, para posteriormente enviarlo por correo electrónico a los presos.

Lista con los presos asignados a las actividades

1. Actividad <nombre> con id <id_actividad>:

Preso <nombre>

Preso <nombre>

.

.

.

Preso <nombre>

2. Actividad <nombre> con id <id_actividad>:

Preso <nombre>

.
.
.

.
.
.

n. Actividad <nombre> con id <id_actividad>:

Preso <nombre>

.
.
.

6.2.7. Diseño y explicación elemento 7

Plantilla del informe de acceso de los Profesores a la prisión

Esta plantilla será la utilizada para generar el informe que posteriormente se enviará por correo electrónico al Jefe del Departamento de Coordinación de Actividades de la prisión para que tenga la información y valide las entradas y salidas de los Profesores a la prisión. La plantilla será la siguiente:

Informe del acceso de los Profesores a la prisión

El profesor <nombre> con id <id_usuario> ha ingresado en el recinto a las <hora_entrada> y ha salido a las <hora_salida>.

El profesor <nombre> con id <id_usuario> ha ingresado en el recinto a las <hora_entrada> y ha salido a las <hora_salida>.

.
.
.

El profesor <nombre> con id <id_usuario> ha ingresado en el recinto a las <hora_entrada> y ha salido a las <hora_salida>.