

ACOES



MEMORIA MODELO E/R Y ENTIDADES JPA

SISTEMAS DE INFORMACIÓN PARA INTERNET



ACOES

Contenido

INTRODUCCIÓN.....	3
Modelo E/R.....	3
Entidades.....	4
Relaciones	4
Entidades JPA	6

MIEMBROS DEL GRUPO

Andrés Valentín Suárez Mediavilla - 77425032V

Sergio González Sicilia - 77225653W

Ignacio Pascual Gutiérrez - 54236255R

Katia Moreno Berrocal - 79038916Z

Rocío Montalvo Lafuente - 03138701Y

URL Repositorio:

<https://github.com/Lilithsoul/ACOES>

Código Hash donde se encuentran las entidades JPA del modelo: 9fd3114

<https://github.com/Lilithsoul/ACOES/commit/9fd311419b74bc30326f75bd8233a7b01f81e9f0>

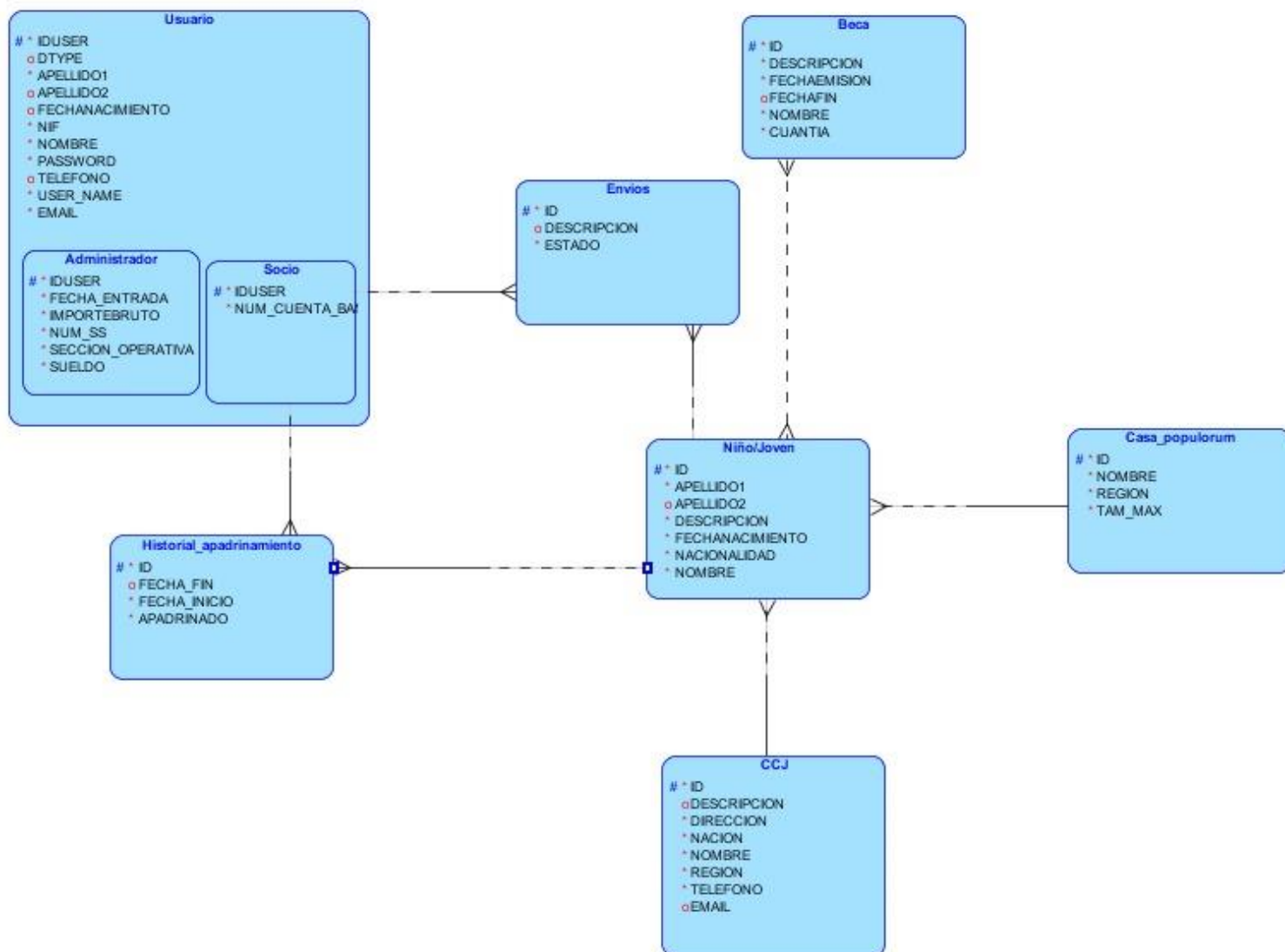


INTRODUCCIÓN.

Para esta entrega hemos realizado un modelo E/R que se ajusta a los requisitos previamente identificados y también se incluyen las entidades JPA correctamente implementadas, las cuales reflejaran los atributos y relaciones presentes en el modelo E/R.

Adjuntamos al final del documento la URL correspondiente al repositorio GitHub donde se encuentra todos los documentos y el proyecto Java.

Modelo E/R





Entidades

Hemos identificado 7 entidades:

- **Usuario:** Compuesto de dos subentidades, socio y administrador. Es necesario establecer esta separación dado que un administrador y un socio poseerán permisos y atributos diferentes. (Con esta entidad se cumple el requisito de registro no siendo un usuario necesariamente un socio).
- **Niño/Joven:** Esta entidad se corresponde con los niños/jóvenes pertenecientes a la organización, y sus atributos serán id, nombre, etc.
- **CCJ:** Esta entidad almacenará los CCJ afiliados a ACOES, junto con su contacto y características básicas. No se tiene en cuenta ningún atributo de calificaciones del niño/joven ni los expedientes. (Con esta entidad se cumple el requisito de creación y administración de CCJ).
- **Casas populum:** Al igual que los CCJ se utilizará para almacenar las casas afiliadas a ACOES, aunque en este caso se almacenará la capacidad máxima que existe en cada de las casas. (Con esta entidad se cumple el requisito de creación y administración de casas populum).
- **Historial de Apadrinamiento:** en esta entidad se almacenarán los niños/jóvenes que estén o hayan sido apadrinados por un usuario, esto se utilizará para que cada uno de los socios pueda ver cuáles son los niños/jóvenes que tiene en ese momento apadrinados o que haya apadrinado con anterioridad. También añadimos un atributo de tipo boolean llamado apadrinado para saber si un niño está siendo apadrinado o no.
- **Beca:** esta entidad almacena los diferentes tipos de becas que existen y que son adjudicadas a los niños/jóvenes. Los administradores son los encargados de crear/modificar... las becas.
- **Envíos:** en esta entidad donde se almacenan los diferentes envíos que realiza un usuario hacia un niño/joven.

Relaciones

Las relaciones que hemos utilizado han sido:

- **Niño/Beca:** Una relación muchos a muchos opcional por ambos lados utilizada para poder relacionar cada una de las becas disponibles con los niños/jóvenes a los que se les ha otorgado. Opcional debido a que un niño/joven puede o no tener una beca y una beca puede no tener un destinatario. (Mediante esta relación se cumple el requisito de gestión de becas).



- **Niño/Casas Populorum:** Relación muchos a uno opcional por parte de los niños/jóvenes ya que un niño no tiene porque estar en una casa populorum, ya que puede encontrarse en un CCJ, esta relación permite saber en qué casa se encuentra cada uno de los niños/jóvenes. (Mediante esta relación se cumple el requisito de asignar niños a una determinada casa).
- **Niño/CCJ:** Relación muchos a uno similar a la de las Casas populorum, que de igual forma se utiliza para almacenar donde se encuentra cada uno de los niños. (Mediante esta relación se cumple el requisito de asignar niños a un determinado CCJ).
- **Niño/Envíos/Socio:** Envíos se encuentra relacionada con niño y socios (subentidad de Usuario) ya que lo que se busca es saber el socio que ha realizado un envío y el niño/joven al que se le destina el envío. Por parte de socio y de niño poseen una relación opcional debido a que un socio puede o no realizar un envío y un niño/joven puede o no recibir un envío. (Mediante esta relación se cumple el requisito en el que se permite a los socios realizar envíos a los niños).
- **Niño/Historial de apadrinamiento/Socio:** Historial de apadrinamiento se encuentra relacionada con socio y niño/joven por relaciones uno a muchos de forma opcional por parte del socio y de niño/joven ya que pueden o no apadrinar o ser apadrinado respectivamente. Esta relación se utiliza para poder almacenar los apadrinamientos realizados o hayan sido realizados por un socio. Un niño solo puede ser apadrinado por un socio. (Mediante esta relación se cumple el requisito de que un usuario pueda apadrinar a uno o mas niños y un niño solo pueda ser apadrinado por un único socio).



Entidades JPA

Las entidades JPA se corresponden con las entidades previamente identificadas en el modelo E/R. Su desarrollo e implementación han sido realizados en el entorno de NetBeans al igual que las prácticas de laboratorio.

En el archivo ***createACOES.sql*** es donde se implementan las tablas y las relaciones, para la explicación lo dividimos en 3 partes:

- ***Creación de tablas:*** en este apartado se crean las tablas principales de nuestro modelo junto a sus atributos (diferenciando los obligatorios de los opcionales con la restricción “NOT NULL”) y estableciendo su clave primaria.
- ***Creación becas.obt:*** en este apartado se crea la tabla becas_obt debido a la relación muchos a muchos establecida entre JOVEN_NIÑO y BECA donde se hereda la clave primaria de cada uno de ellos
- ***Restricciones clave foránea:*** en este apartado se establecen las claves foráneas, las cuales se heredan debido a las relaciones uno a muchos (1:M) establecidas entre las entidades del modelo propuesto.

Estas entidades se encuentran implementadas en el repositorio de GitHub, las cuales generaran el esquema de la base de datos al ejecutar la clase principal.