

# Modelo de Datos

KIUSH

Bahamonde Yohana, Gleadell Carla  
Yield Yields



## Tabla de contenido

<b>Introducción .....</b>	<b>3</b>
<i>Propósito .....</i>	<i>3</i>
<i>Alcance .....</i>	<i>3</i>
<b>Referencias.....</b>	<b>3</b>
<b>Diseño de Datos .....</b>	<b>4</b>
<i>Modelo de Datos General .....</i>	<i>5</i>
<i>Modelo Entidad-Relación .....</i>	<i>5</i>
Definición .....	5
<i>Descripción de Entidades y Atributos .....</i>	<i>6</i>
Llaves .....	6
<i>Relaciones Encontradas .....</i>	<i>7</i>
<i>Diagrama E-R .....</i>	<i>8</i>
Principios de diseño .....	8
Notación diagrama E-R .....	9
<i>Modelo Relacional.....</i>	<i>10</i>
<i>Normalización .....</i>	<i>10</i>
<i>Justificación de Forma normal adoptada.....</i>	<i>11</i>
<i>Especificación de la Distribución de Datos .....</i>	<i>11</i>

# Modelo de Datos

---

## Introducción

Este documento presenta el modelo de datos que se ha desarrollado para la aplicación KIUSH. El modelo de datos es una representación conceptual de los datos que se utilizan en nuestra aplicación, y proporciona una visión detallada de como se organizan, relacionan y manipulan estos datos.

## Propósito

El propósito de este documento es poder proporcionar una perspectiva general del modelo de datos, la cual será utilizada por los desarrolladores para tener en cuenta los datos que se manejará, los nombres de las variables y la estructura de la base de datos, lo que permitirá alcanzar los siguientes objetivos:

- Registrar los requerimientos
- Registrar el modelo de diseño
- Observar usos potenciales de los datos.

## Alcance

Este documento influye directamente en cómo se va a llevar a cabo la implementación del producto KIUSH, ya definirá el total de los datos que se almacenarán y se manipularán en el sistema.

## Referencias

Los documentos involucrados en este Modelo de Datos son:

- Especificación de Requerimientos
- Modelo de Casos de Uso
- Arquitectura del Sistema

## Diseño de Datos

### Entidades

Usuario: usuarios administrativos que pueden gestionar cursos desde el área de extensión de la UNPA-UARG. Integra con sistema UARGFlow.

Rol: define los diferentes niveles de acceso en el sistema (administrador, gestor de cursos, etc.).

Permiso: especifica las acciones que puede realizar cada rol en el sistema.

Curso: representa los cursos de extensión aprobados por la secretaría.

Persona: datos de todas las personas que interactúan con el sistema (inscriptos y docentes).

Integrante: personas inscritas en cursos específicos.

Tipo: categoriza diferentes elementos del sistema (tipos de persona, estados, etc.).

Carrera: carreras académicas de la UNPA-UARG.

Dirección: información de ubicación geográfica.

### Atributos:

Usuario: id, nombre, email (integración UARGFlow)

Rol: id, nombre, descripción

Permiso: id, nombre, descripción

Curso: id, nombre, descripción, fechaInicioInscripcion, fechaFinInscripcion, fechaInicio, fechaFin, horasActividad, lugar, estado, imagen, usuario\_id

Persona: id, nombre, apellido, email, dni, tipo\_id, carrera\_cod

Integrante: id, nombres, apellidos, dni, titulo, instituto, categoriaDocente, dedicacion, direccion, localidad, provincia, codPostal, telefono, email, organizacion, funcion, nvEstudios, seccionTrabEspeciales, tipo\_id

Tipo: id, nombre

Carrera: cod, nombre

Dirección: pais, provincia, localidad, codPostal

### Relaciones

- Usuario-Rol: Un usuario puede tener múltiples roles (n:n)

- Rol-Permiso: Un rol puede tener múltiples permisos (n:n)

- Usuario-Curso: Un usuario puede crear múltiples cursos (1:n)

- Curso-Persona: Un curso puede tener múltiples personas asociadas (n:n)
- Curso-Integrante: Un curso puede tener múltiples integrantes (n:n)
- Persona-Tipo: Una persona tiene un tipo específico (n:1)
- Persona-Carrera: Una persona puede estar asociada a una carrera (n:1)
- Integrante-Tipo: Un integrante tiene un tipo específico (n:1)

### Reglas y restricciones

Un inscripto puede anotarse a un curso solo si el estado del curso es disponible.

## Modelo de Datos General

No existen datos no pertenecientes al núcleo ya que la base de datos UARGFlow otorgada por los docentes fue incorporada dentro del sistema ya que el sistema requiere un registro de los usuarios y esta entidad usuario esta contemplada en UARGFlow, por lo cual no existen dependencias.

## Modelo Entidad-Relación

### Definición

El modelo debe estar compuesto por:

- Entidad: Una entidad es cualquier objeto (sea real o sea abstracto) que existe en el mundo y se quiere almacenar en una base de datos. La representación gráfica de una entidad en este modelo es a través de un rectángulo.
- Atributo: Un atributo es algo que caracteriza a una entidad o a una relación, y una entidad o relación puede estar caracterizada por múltiples atributos. La representación gráfica de un atributo en este modelo es a través de un círculo con el nombre del atributo unido a la entidad de la que es atributo.
- Relación: Una relación es una vinculación entre dos o más entidades. La representación gráfica de una relación en este modelo es a través de un rombo etiquetado con el nombre de la relación unido a las entidades que participan de dicha relación mediante rectas/líneas.
- Cardinalidad: La cardinalidad representa a ambos lados de una relación el "grado" de participación que tiene determinada entidad en esa relación.
- Llave: conjunto de uno o más atributos que "juntos" identifican de manera única a una entidad, la representación gráfica de una llave en este modelo es a través de un subrayado en el atributo que conforma la llave.

## Descripción de Entidades y Atributos

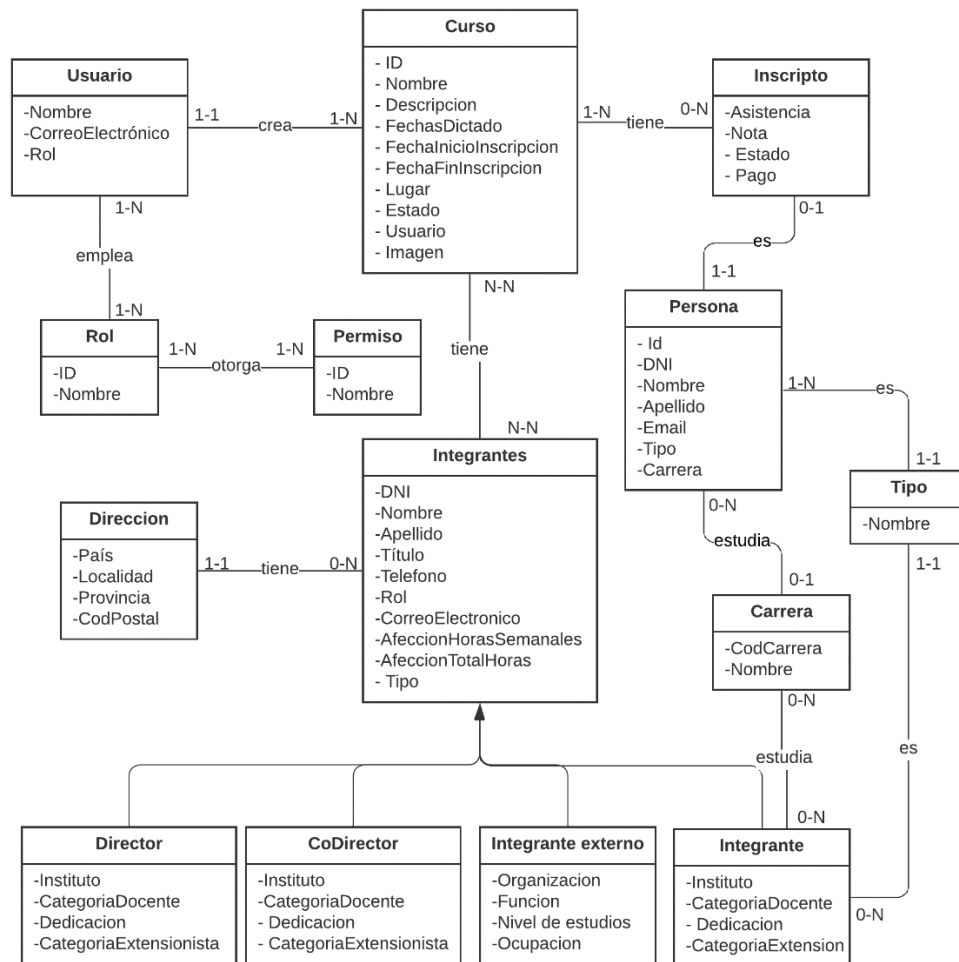


Diagrama de Clases

### Llaves

#### Entidades

Usuario: Llave primaria: id

Rol: Llave primaria: id

Permiso: Llave primaria: id

Curso: Llave primaria: id

Persona: Llave primaria: id

Integrante: Llave primaria: id

Tipo: Llave primaria: id

Carrera: Llave primaria: cod

Dirección: Llave primaria: codPostal

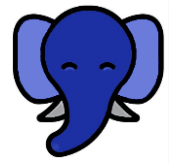
#### Tablas de relación:

Usuario\_rol: id\_usuario, id\_rol

Rol\_permiso: id\_rol, id\_permiso

Curso\_persona: curso\_id, persona\_id

Curso\_integrante: curso\_id, integrante\_id



## Relaciones Encontradas

### Relación 1:

- Entidades relacionadas: usuario - rol
- Nombre: usuario\_rol
- Cardinalidad: n-n

### Relación 2:

- Entidades relacionadas: rol - permiso
- Nombre: rol\_permiso
- Cardinalidad: n-n

### Relación 3:

- Entidades relacionadas: usuario - curso
- Nombre: crea
- Cardinalidad: 1-n

### Relación 4:

- Entidades relacionadas: curso -persona
- Nombre: curso\_persona
- Cardinalidad: n-n

### Relación 5:

- Entidades relacionadas: curso - integrante
- Nombre: curso - integrante
- Cardinalidad: n-n

### Relación 6:

- Entidades relacionadas: persona-tipo
- Nombre: es
- Cardinalidad: n-1

### Relación 7:

- Entidades relacionadas: persona-carrera
- Nombre: estudia
- Cardinalidad: n-1

### Relación 8:

- Entidades relacionadas: integrante-tipo
- Nombre: es
- Cardinalidad: n-1

## Diagrama E-R

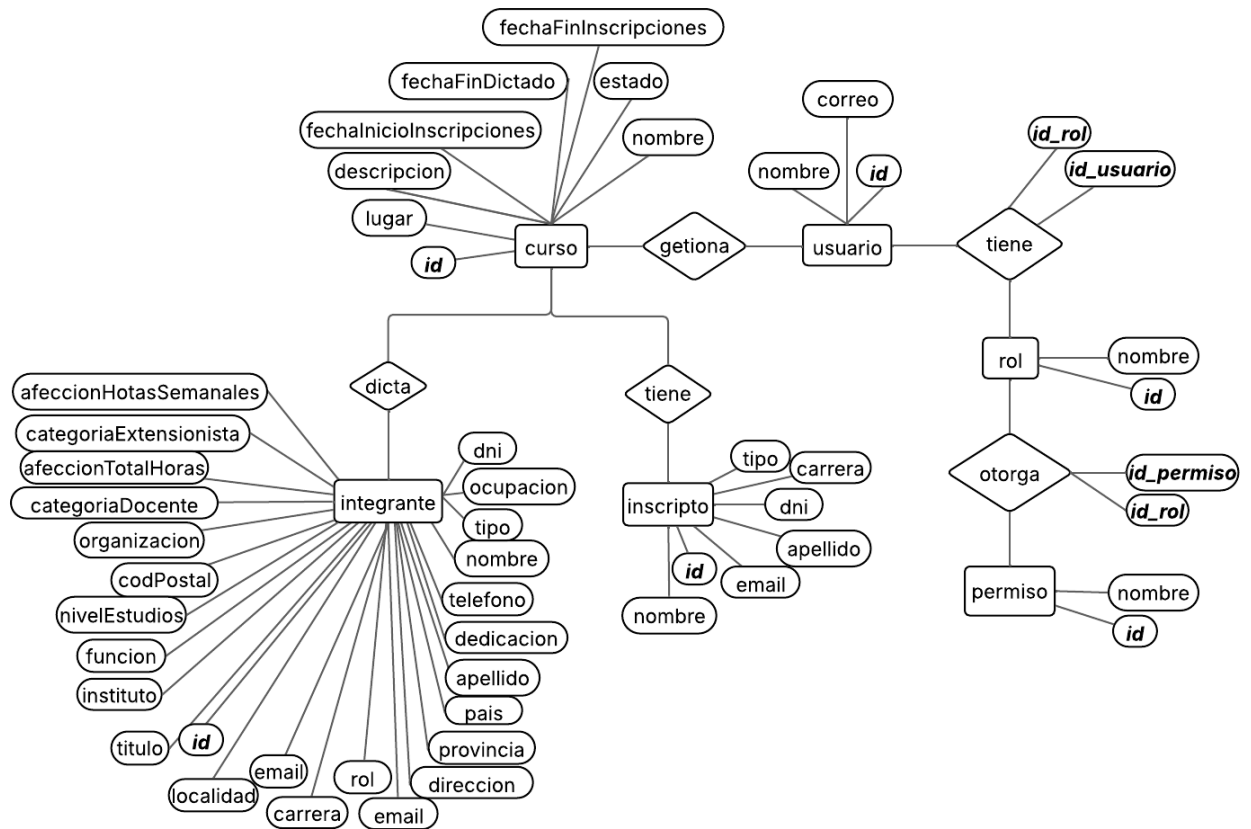


Diagrama Entidad - Relación

## Principios de diseño

**Fidelidad:** se debe crear siempre un modelo que satisfaga las necesidades del problema, no sirve un modelo correcto si no cumple con la realidad que se pretende representar.

**Evitar redundancia:** una de las ventajas del diagrama e-r es que nos permite distinguir de una manera fácil y visual todos los entes y sus relaciones, de manera que es muy fácil identificar si un atributo se está repitiendo en varias entidades o si una relación es innecesaria.










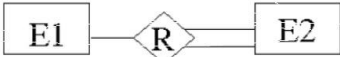


**Simplicidad:** siempre hay que procurar hacer el modelo tan simple como sea posible (sin olvidar la fidelidad) de manera que sea fácil de entender, fácil de extender y fácil de implementar.

**Escoger los elementos correctos:** es ocasiones es difícil identificar si una relación, elemento o atributo es correcto, para ello hay que analizar en perspectiva el diagrama y, por ejemplo, si se observa una entidad con solo un atributo y que únicamente presenta relaciones de 1, entonces probablemente estamos hablando de un atributo y no de una entidad.



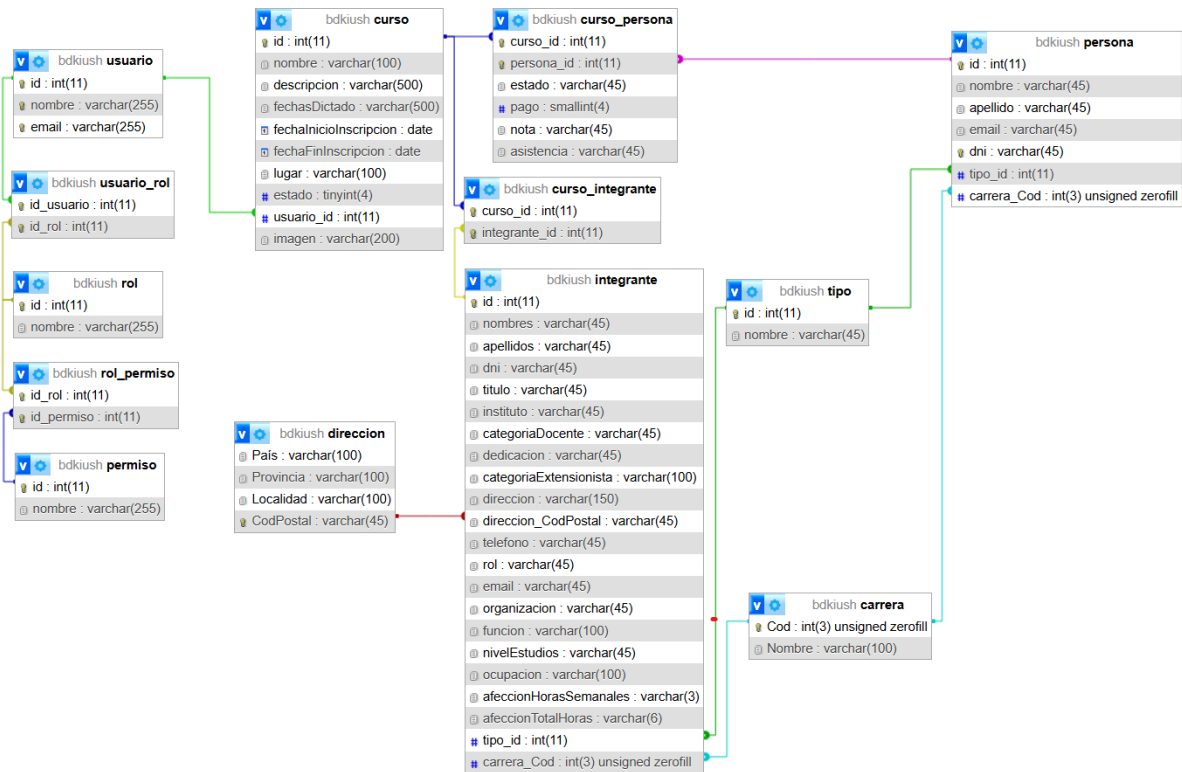
**Relaciones n-arias:** Aun cuando se pueden presentar casos en los que una relación terciaria o n-aria parezca más conveniente, es mejor siempre pensar en términos de relaciones binarias únicamente. En el peor de los casos de que exista una relación n-aria forzosa, lo que se debe hacer es convertir esa relación R en entidad E y corregir todas las relaciones que tenía R de manera que ahora esa nueva entidad se relacione con todas las entidades que anteriormente esta.

## Notación diagrama E-R

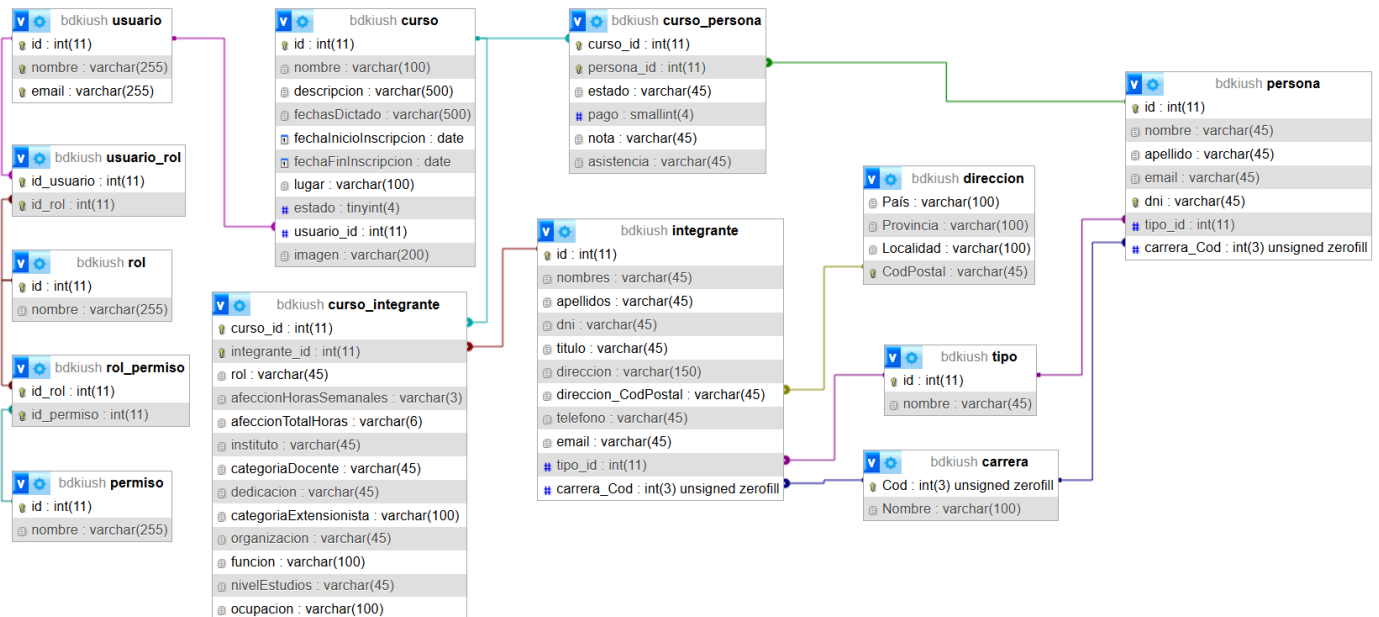
Símbolo	Significado
	TIPO ENTIDAD
	TIPO ENTIDAD DÉBIL
	TIPO VÍNCULO
	VÍNCULO IDENTIFICADOR
	ATRIBUTO
	ATRIBUTO CLAVE
	ATRIBUTO MULTIVALUADO
	ATRIBUTO COMPUESTO
	ATRIBUTO DERIVADO
	PARTICIPACIÓN TOTAL DE E2 EN R
	RAZÓN DE CARDINALIDAD 1:N PARA E1:E2 EN R
	RESTRICCIÓN ESTRUCTURAL (MÍN, MÁX) DE LA PARTICIPACIÓN DE E EN R

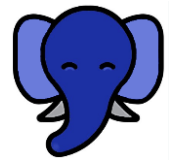
## Notación Diagrama E-R

## Modelo Relacional



## Normalización





## Justificación de Forma normal adoptada

El modelo de datos KIUSH adopta la Tercera Forma Normal (3NF) por las siguientes razones:

- Eliminación de redundancia: Cada tabla contiene información específica sin duplicación innecesaria. Por ejemplo, los datos de usuario están separados de los roles, evitando repetir información de usuario en cada asignación de rol.
- Integridad referencial: Las relaciones entre tablas están claramente definidas mediante claves foráneas, garantizando consistencia de datos. Un curso siempre referencia a un usuario válido que lo creó.
- Flexibilidad del sistema: La separación de entidades permite modificaciones independientes. Se pueden agregar nuevos roles o permisos sin afectar la estructura de usuarios.
- Optimización de consultas: La estructura normalizada permite consultas eficientes y actualizaciones consistentes.
- Escalabilidad: El diseño permite agregar nuevas funcionalidades manteniendo la integridad del modelo.

### Beneficios implementados:

- Consistencia de datos garantizada
- Eliminación de anomalías de inserción, actualización y eliminación
- Mantenimiento simplificado del modelo
- Reutilización eficiente de datos relacionales

## Especificación de la Distribución de Datos

### Ubicación física:

- Servidor principal: Base de datos MariaDB hospedada en servidor local XAMPP
- Manejador: MariaDB 10.4.32
- Servidor web: Apache 2.4.58 con PHP 8.2.12
- Esquema principal: bdklush
- Interfaz de administración: PHPMYAdmin 5.2.1

### Bases de datos:

- bdklush (principal): Contiene todas las entidades del sistema
- Integración con UARGFlow para autenticación

### Índices implementados:

- Claves primarias en todas las tablas principales
- Índices en claves foráneas para optimizar consultas