



Persistencia de datos del Portafolio Móvil

“Presenta tu talento, en cualquier lugar y momento”

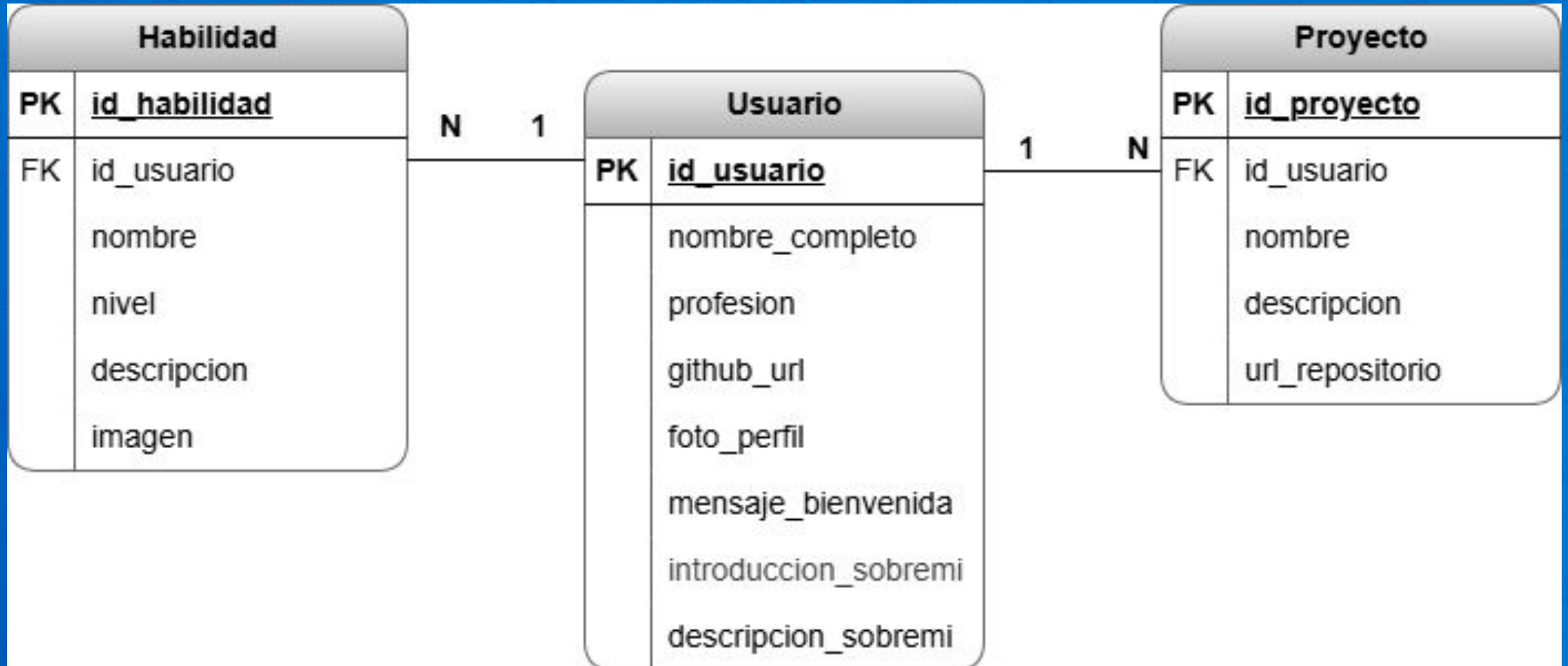
Estudiantes: Sarita Marinao, Juan Cornejo
Profesor: Cristian Iglesias

Temuco · 09/10/2025



UNIVERSIDAD
CATÓLICA DE
TEMUCO

Entidades y relaciones identificadas



Modelo Relacional

1. Tabla: `usuario`

Columna	Tipo de dato	Restricciones	Comentarios
-----	-----	-----	-----
`id_usuario`	INTEGER	PRIMARY KEY, AUTOINCREMENT	Identificador único del usuario.
`nombre_completo`	TEXT	NOT NULL	Nombre completo del usuario.
`profesion`	TEXT	NOT NULL	Profesión del usuario.
`github_url`	TEXT	NOT NULL	URL de GitHub del usuario.
`foto_perfil`	TEXT	NOT NULL	Ruta de la imagen de perfil del usuario.
`mensaje_bienvenida`	TEXT	NOT NULL	Mensaje de bienvenida en el inicio.
`introduccion`	TEXT	NOT NULL	Introducción breve sobre el usuario.
`descripcion`	TEXT	NOT NULL	Descripción detallada sobre el usuario.

****Cardinalidad****: Un usuario puede tener solo un conjunto de datos, por lo que la relación con las tablas `proyectos` y `habilidades` es ****1:1****.

Modelo Relacional

2. Tabla: `proyectos`

Columna	Tipo de dato	Restricciones	Comentarios
-----	-----	-----	-----
`id_proyecto`	INTEGER	PRIMARY KEY, AUTOINCREMENT	Identificador único del proyecto.
`id_usuario`	INTEGER	FOREIGN KEY	Referencia al `id_usuario` de la tabla `usuario`.
`nombre`	TEXT	NOT NULL	Nombre del proyecto.
`descripcion`	TEXT	NOT NULL	Descripción del proyecto.
`url_repositorio`	TEXT	NOT NULL	URL del repositorio del proyecto.
`imagen`	TEXT	NOT NULL	Ruta de la imagen asociada al proyecto.

Cardinalidad: Un usuario puede tener varios proyectos, por lo que la relación con la tabla `usuario` es **1:N** (uno a muchos).



UNIVERSIDAD
CATÓLICA DE
TEMUCO

Modelo Relacional

3. Tabla: `habilidades`

Columna	Tipo de dato	Restricciones	Comentarios
-----	-----	-----	-----
`id_habilidad`	INTEGER	PRIMARY KEY, AUTOINCREMENT	Identificador único de la habilidad.
`id_usuario`	INTEGER	FOREIGN KEY	Referencia al `id_usuario` de la tabla `usuario`.
`nombre`	TEXT	NOT NULL	Nombre de la habilidad (por ejemplo, "JavaScript").
`nivel`	TEXT	NOT NULL	Nivel de la habilidad (por ejemplo, "Avanzado").
`descripcion`	TEXT	NOT NULL	Descripción detallada sobre la habilidad.
`imagen`	TEXT	NOT NULL	Ruta de la imagen asociada a la habilidad.

****Cardinalidad****: Un usuario puede tener varias habilidades, por lo que la relación con la tabla `usuario` es ****1:N**** (uno a muchos).

Modelo Relacional

Relación entre las tablas

1. **Usuario - Proyectos**: La relación es de **1:N**. Un usuario puede tener múltiples proyectos registrados en la tabla `proyectos``.
2. **Usuario - Habilidades**: La relación es de **1:N**. Un usuario puede tener múltiples habilidades en la tabla `habilidades``.

Decisiones de diseño

- Se utilizó `TEXT` para almacenar cadenas de texto, como nombres, descripciones y URLs, ya que se espera que sean datos de longitud variable.
- Las claves foráneas (`id_usuario`) aseguran la integridad referencial entre las tablas. Esto significa que no pueden existir registros en las tablas `proyectos`` y `habilidades`` sin un usuario válido en la tabla `usuario``.
- Los identificadores de las tablas están definidos como `INTEGER PRIMARY KEY AUTOINCREMENT`, lo que significa que serán generados automáticamente por SQLite.

Esquema SQL (schema.sql)

```
-- Crear la tabla "usuario"
CREATE TABLE IF NOT EXISTS usuario (
  id_usuario INTEGER PRIMARY KEY AUTOINCREMENT,
  nombre_completo TEXT NOT NULL,
  profesion TEXT NOT NULL,
  github_url TEXT NOT NULL,
  foto_perfil TEXT NOT NULL,
  mensaje_bienvenida TEXT NOT NULL,
  introduccion TEXT NOT NULL,
  descripcion TEXT NOT NULL
);

-- Crear la tabla "proyectos"
CREATE TABLE IF NOT EXISTS proyectos (
  id_proyecto INTEGER PRIMARY KEY AUTOINCREMENT,
  id_usuario INTEGER,
  nombre TEXT NOT NULL,
  descripcion TEXT NOT NULL,
  url_repositorio TEXT NOT NULL,
  imagen TEXT NOT NULL,
  FOREIGN KEY (id_usuario) REFERENCES usuario (id_usuario) ON DELETE CASCADE
);

-- Crear la tabla "habilidades"
CREATE TABLE IF NOT EXISTS habilidades (
  id_habilidad INTEGER PRIMARY KEY AUTOINCREMENT,
  id_usuario INTEGER,
  nombre TEXT NOT NULL,
  nivel TEXT NOT NULL,
  descripcion TEXT NOT NULL,
  imagen TEXT NOT NULL,
  FOREIGN KEY (id_usuario) REFERENCES usuario (id_usuario) ON DELETE CASCADE
);
```

Mapping de Funcionalidades

Mapping de funcionalidades

Funcionalidad / Pantalla	Tablas / Datos (SQLite)	Archivos	Descripción
Inicio / Sobre mí	usuario	main.py	Muestra y permite editar información personal del usuario.
Listado de proyectos	proyectos	main.py	Visualiza proyectos almacenados en la base de datos.
Crear / Editar proyecto	proyectos	main.py	Permite añadir o modificar proyectos existentes.
Habilidades	habilidades	main.py	Muestra las habilidades con sus niveles correspondientes.
Persistencia (CRUD)	usuario, proyectos, habilidades	gestor.py	Funciones de conexión y consultas SQLite.
Navegación	—	portafolio.kv	Gestión de pantallas mediante Drawer o ScreenManager.
Estilos / Diseño	—	portafolio.kv	Definición de colores, tipografía y layout general de la app.

Plan de implementación

Fase / Tarea	Responsable	Duración (horas)	Inicio	Término	Estado	Objetivo / Entregable
Fase 1: Preparación						
Crear estructura base de main.py	Juan Cornejo	2	07/10	07/10	Completado <input checked="" type="checkbox"/>	Proyecto base ejecutable.
Configurar entorno y dependencias (Kivy, KivyMD)	Juan Cornejo	1	07/10	07/10	Completado <input checked="" type="checkbox"/>	Entorno de desarrollo funcional.
Fase 2: Base de datos y persistencia						
Crear base SQLite y tablas (usuario, proyectos, habilidades)	?	3	08/10	08/10	En progreso <input type="checkbox"/>	Archivo proyectos.db generado correctamente.
Integrar funciones CRUD básicas	?	4	09/10	09/10	Pendiente <input type="checkbox"/>	CRUD operativo en consola o app.
Fase 3: Interfaz y navegación						
Diseñar pantalla de inicio y menú de navegación	Juan Cornejo	4	10/10	10/10	Completado <input checked="" type="checkbox"/>	Navegación entre secciones funcional.
Conectar botones del menú con pantallas	Juan Cornejo	3	11/10	11/10	Completado <input checked="" type="checkbox"/>	Menú enlazado correctamente.
Fase 4: Funcionalidades de contenido						
Listar proyectos desde la base	?	5	12/10	13/10	Pendiente <input type="checkbox"/>	Lista funcional con scroll.
Crear/editar proyectos (formulario)	?	5	14/10	15/10	Pendiente <input type="checkbox"/>	Datos persistentes en SQLite.
Mostrar lista de habilidades	Ambos	3	16/10	16/10	Completado <input checked="" type="checkbox"/>	Habilidades visibles y ordenadas.
Fase 5: Pruebas y documentación						
Pruebas funcionales y corrección de errores	Ambos	4	17/10	17/10	Pendiente <input type="checkbox"/>	App estable y sin fallos.
Redacción de documentación (README, roles, plan)	Ambos	3	18/10	18/10	En progreso <input type="checkbox"/>	Entrega lista para presentación.

Ejemplo Persistencia con Demo