



PONTIFICIA UNIVERSIDAD CATÓLICA DE CHILE  
DEPARTAMENTO DE CIENCIA DE LA COMPUTACIÓN  
IIC2413 - BASES DE DATOS

# Entrega 02

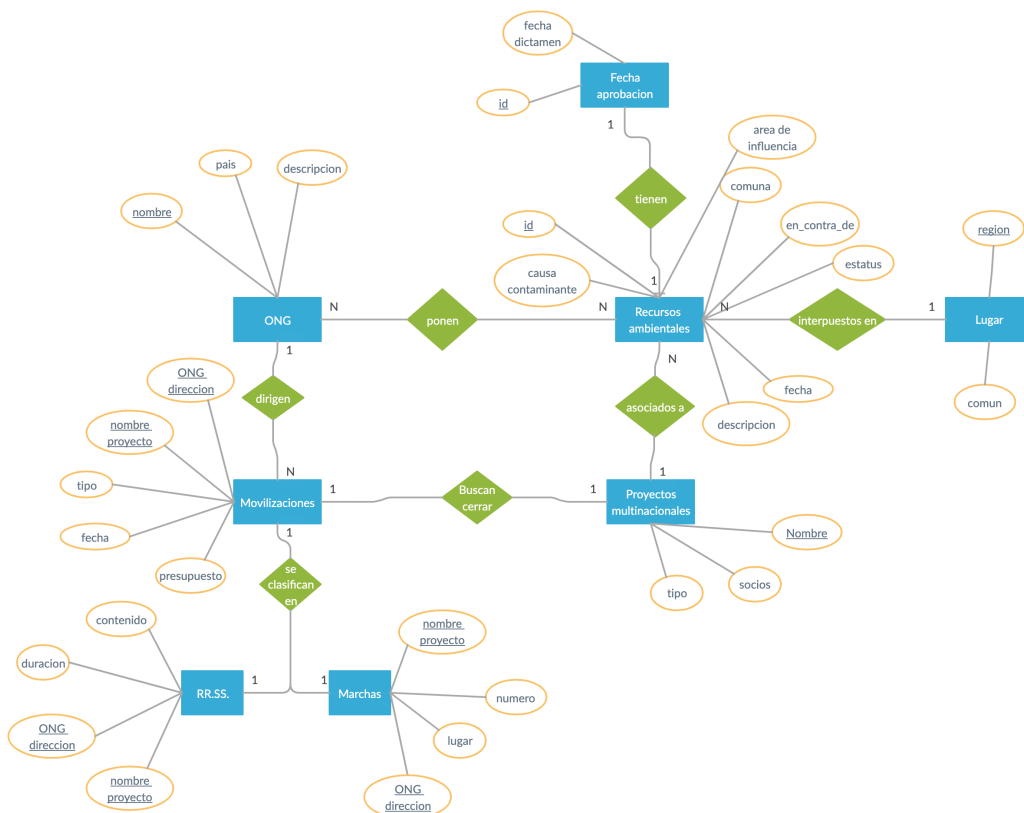
17 de septiembre de 2019

2º semestre 2019 - Profesores J. Reutter - V. Domínguez

Manuel Muñoz Arriagada (16639537) - Joaquín Terreros Salazar (16205642)

## Parte 1: Diagrama entidad-relación

A continuación se presenta el diagrama entidad-relación que se utilizó en primer lugar para entender el problema:



## Parte 2: Esquema relacional

A continuación se presenta el esquema relacional entidad-relación utilizado posteriormente que se subiera la base de datos al servidor (y con la explicación de los atributos). Todos los atributos con **FK** hacen referencia a una *foreign key* que muestra la relación que existe entre dichas tablas (que contienen el mismo atributo):

- FechaAprobación(idr: string **FK**, fecha dictamen: int)
- Lugar(region: string, comuna: string)
- Marchas(nombre proyecto: string **FK**, ONG dirección: string **FK**, lugar: string, numero: float)
- RRSS(nombre proyecto: string **FK**, ONG dirección: string **FK**, duración: float, contenido: string)
- Movilizaciones(nombre proyecto: string **FK**, ONG dirección: string **FK**, tipo: string, fecha: string, presupuesto: float)
- ProyectosMultinacionales(nombre: string, socios: string, tipo: string)
- ONG(nombre: string, pais: string, descripcion: string)
- RecursosAmbientales(idr: string **FK**, nombre proyecto: string **FK**, causa contaminante: string, comuna: string, area de influencia: string, descripcion: string, fecha: string, comuna: string, estatus: string)
- RRSS(idr: string **FK**, ONG participantes: string)

## Parte 3: Justificación 3NF o BCNF

El esquema/diagrama anterior cumple con las condiciones 3NF por:

1. Se encuentra en la cantidad mínima de llaves que permitan identificar un elemento de otro dentro de una misma tabla (o en otras palabras, los atributos llaves son la menor cantidad posible).
2. No existen relaciones redundantes entre las tablas, salvo las foreign keys que relacionan según la modelación, lo cual impide un ciclo en cada consulta.

## Parte 4: Comandos SQL para tablas

Los comandos utilizados para trabajar con las tablas en la base de datos fueron:

1. **CREATE**: comando que permite crear un elemento dentro de la base de datos. En este caso en particular, se usó para crear las tablas con las que haríamos las consultas.
2. **SELECT**: comando que permite seleccionar de una o más tablas para poder filtrar según los requerimientos.
3. **FROM**: comando que identifica la tabla con la cual trabajaremos.
4. **ON**: comando que permite agregar una condición al filtro u operación que se le está aplicando a la tabla.
5. **WHERE**: comando que aplica filtro por aquello que estamos deseando obtener de la tabla.
6. **LIKE**: comando que 'suaviza' el criterio de búsqueda, en caso de que hayan resultados que difieran mínimamente, sean reconocidos igual.

## Parte 5: Consultas en SQL

Las consultas aplicadas para poder realizar las búsquedas (disponibles también en la base de datos, carpeta consultas) son las siguientes:

## Parte 6: Información adicional

La planilla utilizada para generar las consultas fue extraída de la ayudantía realizada una semanas antes