Spring Data: CRUD repositories

```java
public interface PersonaRepository extends Repository<Persona, Long> {

  public List<Persona> findByNom(String nom);

}
```

CRUD = Create, Read, Update i Delete

CrudRepository

```
public interface PersonaRepository extends CrudRepository<Persona, Long> {

  public List<Persona> findByNom(String nom);

}
```

CrudRepository = Repository +

- save (entity)
- saveAll (iterable)
- findById (id)
- count()
- delete (entity)

...

Create/
Update

Read

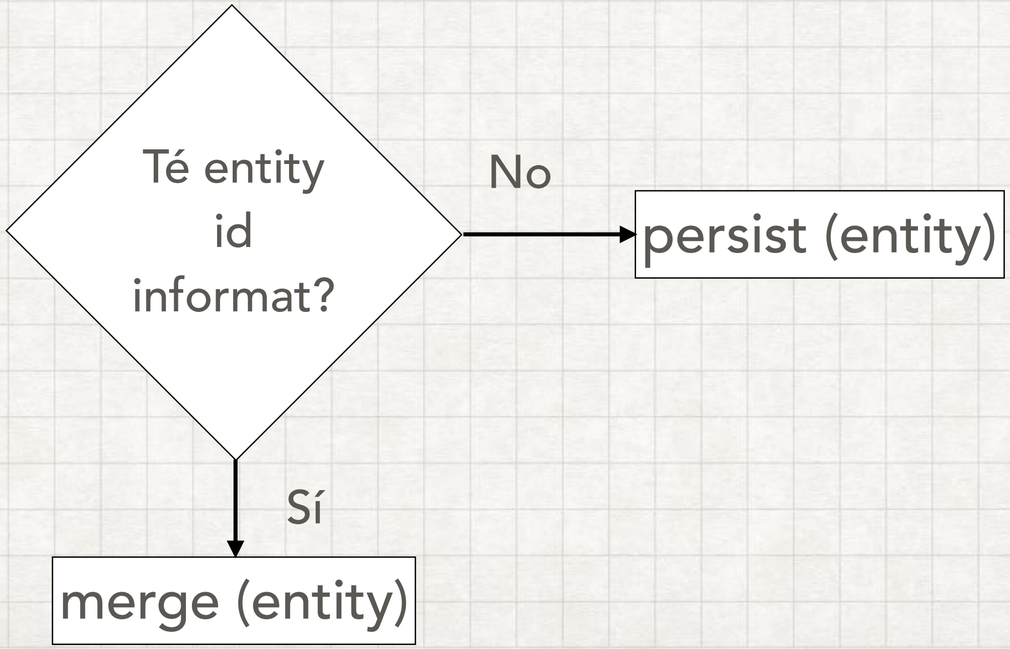
Delete

```
public interface CrudRepository<T, ID> extends Repository<T, ID> {

  <S extends T> S save(S entity);
  <S extends T> Iterable<S> saveAll(Iterable<S> entities);

  Optional<T> findById(ID id);
  boolean existsById(ID id);
  Iterable<T> findAll();
  Iterable<T> findAllById(Iterable<ID> ids);
  long count();

  void deleteById(ID id);
  void delete(T entity);
  void deleteAll(Iterable<? extends T> entities);
  void deleteAll();
}
```

Create/
Update

```
public interface CrudRepository<T, ID> extends Repository<T, ID> {

  <S extends T> S save(S entity);
  <S extends T> Iterable<S> saveAll(Iterable<S> entities);
  ...
}
```

```
S save(S entity);
```



Read

```
public interface CrudRepository<T, ID> extends Repository<T, ID> {
  ...
  Optional<T> findById(ID id);
  boolean existsById(ID id);
  Iterable<T> findAll();
  Iterable<T> findAllById(Iterable<ID> ids);
  long count();
  ...
}
```

Optional<T>

```
Optional<T> retorn = ...;

boolean existeix = retorn.isPresent();
T valor = retorn.get();
T valor = retorn.orElse(altre);
```

Delete

```
public interface CrudRepository<T, ID> extends Repository<T, ID> {
  ...
  void deleteById(ID id);
  void delete(T entity);
  void deleteAll(Iterable<? extends T> entities);
  void deleteAll();
}
```