

Package `com.mapmyjourney.backend.repository`

## Interface `UserRepository`

All Superinterfaces:

`org.springframework.data.repository.CrudRepository<User, Long↗>`,  
`org.springframework.data.jpa.repository.JpaRepository<User, Long↗>`,  
`org.springframework.data.repository.ListCrudRepository<User, Long↗>`,  
`org.springframework.data.repository.ListPagingAndSortingRepository<User, Long↗>`,  
`org.springframework.data.repository.PagingAndSortingRepository<User, Long↗>`,  
`org.springframework.data.repository.query.QueryByExampleExecutor<User>`,  
`org.springframework.data.repository.Repository<User, Long↗>`

@Repository  
public interface **UserRepository**  
extends `org.springframework.data.jpa.repository.JpaRepository<User, Long↗>`

### Method Summary

- All Methods
- Instance Methods
- Abstract Methods

| Modifier and Type  |  |
|--|--|
| Method   |  |
| Description  |  |
| boolean  |  |
| <b>existsByEmail</b> (String <sup>↗</sup> email)   |  |
| Verifica si existe un usuario con ese email.   |  |
| Optional <sup>↗</sup> <User>   |  |
| <b>findByEmail</b> (String <sup>↗</sup> email)   |  |
| Busca un usuario por su email.   |  |
| Methods inherited from interface org.springframework.data.repository.CrudRepository  |  |
| count, delete, deleteAll, deleteAll, deleteAllById, deleteById, existsById, findById, save   |  |
| Methods inherited from interface org.springframework.data.jpa.repository.JpaRepository   |  |
| deleteAllByIdInBatch, deleteAllInBatch, deleteAllInBatch, deleteInBatch, findAll, findAll, flush, getById, getOne, getReferenceById, saveAllAndFlush, saveAndFlush |  |
| Methods inherited from interface org.springframework.data.repository.ListCrudRepository  |  |
| findAll, findAllById, saveAll  |  |
| Methods inherited from interface org.springframework.data.repository.ListPagingAndSortingRepository  |  |
| findAll  |  |
| Methods inherited from interface org.springframework.data.repository.PagingAndSortingRepository  |  |
| findAll  |  |
| Methods inherited from interface org.springframework.data.repository.query.QueryByExampleExecutor  |  |
| count, exists, findAll, findBy, findOne  |  |

### Method Details

|                    |
|--------------------|
| <b>findByEmail</b> |
|--------------------|

`Optional<User> findByEmail(String email)`

Busca un usuario por su email.

**Parameters:**

email - el email del usuario

**Returns:**

Optional con el usuario si existe

## existsByEmail

`boolean existsByEmail(String email)`

Verifica si existe un usuario con ese email.

**Parameters:**

email - el email a verificar

**Returns:**

true si existe, false si no