

criteria usando JPA

todas las personas:

```
EntityManagerFactory emf = Persistence.createEntityManagerFactory(persistenceUnitName:"SgaPU");
EntityManager em = emf.createEntityManager();

CriteriaBuilder cb = null;
CriteriaQuery<Persona> criteriaQuery = null;
Root<Persona> fromPersona = null;
TypedQuery<Persona> query = null;
Persona persona = null;
List<Persona> personas = null;

//Query utilizando el API de Criteria
//1. Consulta de todas las personas

//Paso1. El objeto EntityManager crea instancia CriteriaBuilder
cb = em.getCriteriaBuilder();

//Paso2. Se crea un objeto CriteriaQuery
criteriaQuery = cb.createQuery(resultClass:Persona.class);

//Paso3. Creamos el objeto raiz de query
fromPersona = criteriaQuery.from(entityClass:Persona.class);

//Paso4. seleccionamos lo necesario del from
criteriaQuery.select(selection:fromPersona);

//Paso5. Creamos el query typeSafe
query = em.createQuery(criteriaQuery);

//Paso6. Ejecutamos la consulta
personas = query.getResultList();

mostrarPersonas(personas);
```

```
21:00:59 [main] DEBUG - Persona{idPersona=1, nombre=Juan, apellido=Perez, email=jperez@mail.com, telefono=22577777}
21:00:59 [main] DEBUG - Persona{idPersona=2, nombre=Karla, apellido=Gomez, email=kgomez@mail.com, telefono=25252525}
21:00:59 [main] DEBUG - Persona{idPersona=3, nombre=Maria, apellido=Gutierrez, email=mgutierrez@mail.com, telefono=88991199}
21:00:59 [main] DEBUG - Persona{idPersona=4, nombre=Pedro, apellido=Luna, email=pedro@gmail.com, telefono=13132153}
21:00:59 [main] DEBUG - Persona{idPersona=6, nombre=Hugo, apellido=Hernandez, email=hhernandez@mail.com, telefono=55778822}
```

```
//2-a. Consulta de la Persona con id = 1
//jql = "select p from Persona p where p.idPersona = 1"
log.debug(message:"\n2-a. Consulta de la Persona con id = 1");
cb = em.getCriteriaBuilder();
criteriaQuery = cb.createQuery(resultClass:Persona.class);
fromPersona = criteriaQuery.from(entityClass:Persona.class);
criteriaQuery.select(selection:fromPersona).where(restriction:cb.equal(x:fromPersona.get(attributeName:"idPersona"), y:1));
persona = em.createQuery(criteriaQuery).getSingleResult();
log.debug(message:persona);
```

```
2-a. Consulta de la Persona con id = 1
[EL Fine]: sql: 2023-11-02 21:02:15.683--ServerSession(554868511)--Connection(1590481849)--SELECT id_persona, APELLIDO, EMAIL, NOMBRE, TELEFONO FROM PERSONA WHERE (id_persona = ?)
bind => [1 parameter bound]
21:02:15 [main] DEBUG - Persona(idPersona=1, nombre=Juan, apellido=Perez, email=jperez@mail.com, telefono=22577777)
```

```

//2-b. Consulta de la Persona con id = 1
log.debug(message:"\n2-b. Consulta de la Persona con id = 1");
cb = em.getCriteriaBuilder();
criteriaQuery = cb.createQuery(resultClass:Persona.class);
fromPersona = criteriaQuery.from(entityClass:Persona.class);
criteriaQuery.select(selection: fromPersona);

//La clase Predicate permite agregar varios criterios dinamicamente
List<Predicate> criterios = new ArrayList<Predicate>();

//Verificamos si tenemos criterios que agregar
Integer idPersonaParam = 1;
ParameterExpression<Integer> parameter = cb.parameter(paramClass: Integer.class, name: "idPersona");
criterios.add( e: cb.equal(n: fromPersona.get(attributeName: "idPersona"), y: parameter));

//Se agregan los criterios
if(criterios.isEmpty()){
    throw new RuntimeException(string: "Sin criterios");
}
else if(criterios.size() == 1){
    criteriaQuery.where(restriction:Criterios.get(index: 0));
}
else{
    criteriaQuery.where(restriction:cb.and(restrictions:criterios.toArray(new Predicate[0])));
}

query = em.createQuery(criteriaQuery);
query.setParameter(name: "idPersona", value: idPersonaParam);

//Se ejecuta el query
persona = query.getSingleResult();
log.debug(message:persona);

```

```

21:02:48 [main] DEBUG - Persona(idPersona=1, nombre=Juan, apellido=Perez, email=jperez@mail.com, telefono=22577777)
[EL Fine]: sql: 2023-11-02 21:02:48.939--ServerSession(554668511)--Connection(1179272258)--SELECT id_persona, APELLIDO, EMAIL, NOMBRE, TELEFONO FROM PERSONA WHERE (id_persona = ?)
bind => [1 parameter bound]

```