



Figure IV.11 – Fonctionnement d'Apex

La figure IV.12 est une capture d'écran présentant la classe Apex « ListProjetController.apxc » avec ses différentes méthodes qui comprend l'instruction de bouclage, l'instruction de flux de contrôle et la requête SOQL.

```
public with sharing class ListProjetController {
    @AuraEnabled
    public static list < Projet__c > fetchProject() {
        list < Projet__c > returnConList = new List < Projet__c > ();
        list < Projet__c > lstCon = [SELECT Name, iconstatus__c, Description_projet__c, date_debut__c, date_fin__c, Etat__c, Ressource__r.Name From Projet__c ORDER BY date_fin__c DESC NULLS LAST LIMIT 50]
        for (Projet__c c: lstCon) {
            returnConList.add(c);
        }
        return returnConList;
    }
    @AuraEnabled
    public static List <Projet__c> loadData(){
        return [SELECT Name, iconstatus__c, Description_projet__c, date_debut__c, date_fin__c, Etat__c, Ressource__r.Name From Projet__c ORDER BY date_fin__c DESC NULLS LAST LIMIT 50];
    }
    @AuraEnabled
    public static List<Projet__c> UpdateRecord(String conid){
        return [SELECT Id, Name, Description_projet__c, date_debut__c, date_fin__c, Etat__c FROM Projet__c WHERE Id =:conid];
    }
    @AuraEnabled
    public static List<Projet__c> showdetail (String conid){
        return [SELECT Id, Name, Description_projet__c, date_debut__c, date_fin__c, Etat__c FROM Projet__c WHERE Id =:conid];
    }
}
```

Figure IV.12 – Interface de Contrôleur Apex

5 Exemples d'interfaces

Lors de la réalisation de notre application, nous n'avons pas adopté une version complète de CRM Salesforce puisqu'elle coûte environ \$500 par mois et augmentent en fonction du nombre d'utilisateurs.

A cet égard, nous avons utilisé la notion de site comme mécanisme technique rapide de Salesforce permettant de créer des sites web publics et des applications directement intégrés