# Proyecto de Planificación de Horticultura

## Introducción

Este proyecto tiene como objetivo desarrollar una aplicación web para la planificación de tareas de horticultura utilizando archivos PDDL (Planning Domain Definition Language). La aplicación permite a los usuarios generar planes basados en dominios y problemas definidos en PDDL y visualizar estos planes de manera clara en una tabla.

## Estructura del Proyecto

El proyecto está organizado en las siguientes carpetas:

* horticultura\_ai\_planning
* horticultura
* horticultura\_page
* planificador

## Descripción de las Carpetas

### horticultura\_ai\_planning

Esta carpeta contiene el entorno virtual y las dependencias necesarias para ejecutar el proyecto. Asegúrate de activar el entorno virtual antes de ejecutar cualquier comando relacionado con el proyecto.

### horticultura

Esta carpeta contiene el proyecto Django principal. Incluye los archivos de configuración y las aplicaciones necesarias para la funcionalidad del proyecto.

### horticultura\_page

Esta carpeta contiene las plantillas HTML y los archivos estáticos necesarios para la interfaz de usuario de la aplicación web.

### planificador

Esta carpeta contiene la lógica principal de la aplicación, incluyendo las vistas, los modelos y los archivos de configuración de URL.

## Documentación del Código

### views.py

Este archivo contiene las vistas principales de la aplicación. Aquí se generan los planes basados en los archivos PDDL y se renderizan las páginas HTML.

from django.shortcuts import render  
from pyperplan.pddl.parser import Parser  
from pyperplan.planner import \_parse, \_ground, \_search  
from pyperplan.search import astar\_search  
from pyperplan.heuristics.blind import BlindHeuristic  
import os  
import pandas as pd  
  
def generar\_plan(request):  
 """  
 Genera un plan basado en los archivos PDDL y lo muestra en una página web.  
  
 Args:  
 request (HttpRequest): La solicitud HTTP.  
  
 Returns:  
 HttpResponse: La respuesta HTTP con el plan generado y la simulación de resultados.  
 """  
 if request.method == 'POST':  
 # Obtener la ruta absoluta del directorio actual  
 base\_dir = os.path.dirname(os.path.dirname(os.path.abspath(\_\_file\_\_)))  
 domain\_file = os.path.join(base\_dir, 'domain.pddl')  
 problem\_file = os.path.join(base\_dir, 'problem.pddl')  
   
 parser = Parser(domain\_file, problem\_file)  
 domain = parser.parse\_domain()  
 problem = parser.parse\_problem(domain)  
   
 task = \_ground(problem)  
 heuristic = BlindHeuristic(task)  
 plan = \_search(task, astar\_search, heuristic)  
   
 plan\_str = '\n'.join(str(action) for action in plan)  
   
 # Simulación de resultados  
 estados = []  
 estado\_actual = {}  
 for action in plan:  
 estados.append(f"Antes de {action}: {estado\_actual.copy()}")  
 # Actualizar estado\_actual según la acción  
 if "plantar" in str(action):  
 planta = str(action).split()[1]  
 estado\_actual[planta] = "plantada"  
 elif "regar" in str(action):  
 planta = str(action).split()[1]  
 estado\_actual[planta] = "regada"  
 elif "fertilizar" in str(action):  
 planta = str(action).split()[1]  
 estado\_actual[planta] = "fertilizada"  
 elif "podar" in str(action):  
 planta = str(action).split()[1]  
 estado\_actual[planta] = "podada"  
 elif "cosechar" in str(action):  
 planta = str(action).split()[1]  
 estado\_actual[planta] = "cosechada"  
 estados.append(f"Después de {action}: {estado\_actual.copy()}")  
   
 # Convertir estados a DataFrame para mostrar en tabla  
 df\_estados = pd.DataFrame(estados, columns=['Estado'])  
   
 return render(request, 'plan.html', {'plan': plan\_str, 'estados': estados, 'df\_estados': df\_estados.to\_html(classes='table table-striped')})  
 return render(request, 'index.html')

### urls.py

Este archivo contiene las rutas de URL para la aplicación.

from django.urls import path  
from . import views  
  
urlpatterns = [  
 path('', views.generar\_plan, name='generar\_plan'),  
]

### problem.pddl

Este archivo define el problema de horticultura en PDDL.

(define (problem horticultura-problema)  
 (:domain horticultura)  
   
 (:objects  
 planta1 planta2 planta3 - planta  
 regadera fertilizante tijeras - herramienta  
 jardin invernadero huerto - lugar  
 )  
   
 (:init  
 (en-lugar planta1 jardin)  
 (en-lugar planta2 invernadero)  
 (en-lugar planta3 huerto)  
 (tiene-herramienta regadera)  
 (tiene-herramienta fertilizante)  
 (tiene-herramienta tijeras)  
 )  
   
 (:goal  
 (and  
 (cosechada planta1)  
 (cosechada planta2)  
 (cosechada planta3)  
 )  
 )  
)

### domain.pddl

Este archivo define el dominio de horticultura en PDDL.

(define (domain horticultura)  
 (:requirements :strips :typing :action-costs)  
   
 (:types  
 planta herramienta lugar  
 )  
   
 (:predicates  
 (en-lugar ?p - planta ?l - lugar)  
 (tiene-herramienta ?h - herramienta)  
 (plantada ?p - planta)  
 (regada ?p - planta)  
 (fertilizada ?p - planta)  
 (podada ?p - planta)  
 (cosechada ?p - planta)  
 )  
   
 (:action plantar  
 :parameters (?p - planta ?l - lugar)  
 :precondition (and (en-lugar ?p ?l))  
 :effect (plantada ?p)  
 :cost 1  
 )  
   
 (:action regar  
 :parameters (?p - planta)  
 :precondition (and (plantada ?p))  
 :effect (regada ?p)  
 :cost 1  
 )  
   
 (:action fertilizar  
 :parameters (?p - planta)  
 :precondition (and (plantada ?p))  
 :effect (fertilizada ?p)  
 :cost 2  
 )  
   
 (:action podar  
 :parameters (?p - planta)  
 :precondition (and (plantada ?p))  
 :effect (podada ?p)  
 :cost 2  
 )  
   
 (:action cosechar  
 :parameters (?p - planta)  
 :precondition (and (plantada ?p) (regada ?p) (fertilizada ?p) (podada ?p))  
 :effect (cosechada ?p)  
 :cost 3  
 )  
)

## Ejecución del Proyecto

Para ejecutar el proyecto, sigue estos pasos:

### 1. Activa el entorno virtual:

source venv/bin/activate # En Linux/Mac  
venv\Scripts\activate # En Windows

### 2. Instala las dependencias

pip install -r requirements.txt

### 3. ejecuta el servidor

cd horticultura\_ai\_planning  
cd horticultura  
python manage.py runserver

### 4. Abre tu navegador y ve a http://127.0.0.1:8000/ para ver la aplicación en acción.