# 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, redirect**

### **import os**

### **import random**

### **from pyperplan.pddl.parser import Parser**

### **from pyperplan.planner import \_ground**

### **from pyperplan.search import astar\_search**

### **from pyperplan.heuristics.blind import BlindHeuristic**

### **from pyperplan.heuristics.lm\_cut import LmCutHeuristic**

### **# Definir una clase Node simple si no está disponible en pyperplan**

### **class Node:**

### **def \_\_init\_\_(*self*, *state*):**

### **self.state = state**

### **def index(*request*):**

### **"""**

### **Muestra la página principal del planificador.**

### **Args:**

### **request (HttpRequest): La solicitud HTTP.**

### **Returns:**

### **HttpResponse: La respuesta HTTP con la página principal del planificador.**

### **"""**

### **if request.method == 'POST':**

### **if 'action' in request.POST and request.POST['action'] == 'generar\_plan':**

### **return redirect('/generar\_plan')**

### **return render(request, 'index.html')**

### **def heuristica(*request*):**

### **"""**

### **Aplica una heurística al problema PDDL y muestra el resultado.**

### **Args:**

### **request (HttpRequest): La solicitud HTTP.**

### **Returns:**

### **HttpResponse: La respuesta HTTP con el resultado de la heurística.**

### **"""**

### **base\_dir = os.path.dirname(os.path.dirname(os.path.abspath(\_\_file\_\_)))**

### **domain\_file = os.path.join(base\_dir, 'domain.pddl')**

### **problem\_files = [**

### **os.path.join(base\_dir, 'problem\_jardin.pddl'),**

### **os.path.join(base\_dir, 'problem\_invernadero.pddl'),**

### **os.path.join(base\_dir, 'problem\_huerto.pddl')**

### **]**

### **heuristica\_content = ""**

### **for i, problem\_file in enumerate(problem\_files):**

### **# Parsear los archivos PDDL**

### **parser = Parser(domain\_file, problem\_file)**

### **domain = parser.parse\_domain()**

### **problem = parser.parse\_problem(domain)**

### **# Generar la tarea**

### **task = \_ground(problem)**

### **# Aplicar la heurística**

### **heuristic = LmCutHeuristic(task)  # Usar la nueva heurística**

### **initial\_node = Node(task.initial\_state)**

### **heuristic\_value = heuristic(initial\_node)**

### **# Convertir el estado inicial y los objetivos a cadenas de texto legibles**

### **initial\_state\_str = "\n".join(sorted(str(state) for state in task.initial\_state))**

### **goal\_str = "\n".join(sorted(str(goal) for goal in task.goals))**

### **# Añadir detalles al contenido de la respuesta**

### **heuristica\_content += (**

### **f"Camino {i+1}:\n"**

### **f"Estado inicial:\n{initial\_state\_str}\n\n"**

### **f"Objetivos:\n{goal\_str}\n\n"**

### **f"Valor heurístico aplicado al estado inicial: {heuristic\_value}\n\n"**

### **)**

### **return render(request, 'heuristica.html', {'heuristica\_content': heuristica\_content})**

### **def mostrar\_dominio\_y\_problemas(*request*):**

### **base\_dir = os.path.dirname(os.path.dirname(os.path.abspath(\_\_file\_\_)))**

### **domain\_file = os.path.join(base\_dir, 'domain.pddl')**

### **problem\_files = {**

### **'jardin': os.path.join(base\_dir, 'problem\_jardin.pddl'),**

### **'invernadero': os.path.join(base\_dir, 'problem\_invernadero.pddl'),**

### **'huerto': os.path.join(base\_dir, 'problem\_huerto.pddl'),**

### **'general': os.path.join(base\_dir, 'problem.pddl')**

### **}**

### **problems = {}**

### **with open(domain\_file, 'r', *encoding*='utf-8') as file:**

### **domain\_content = file.read()**

### **for problem\_type, problem\_file in problem\_files.items():**

### **with open(problem\_file, 'r', *encoding*='utf-8') as file:**

### **problems[problem\_type] = file.read()**

### **return render(request, 'dominio\_y\_problemas.html', {'domain\_content': domain\_content, 'problems': problems})**

### **def generar\_plan(*request*):**

### **"""**

### **Genera un plan para cada uno de los problemas específicos.**

### **Args:**

### **request (HttpRequest): La solicitud HTTP.**

### **Returns:**

### **HttpResponse: Redirige a la página de ver plan.**

### **"""**

### **base\_dir = os.path.dirname(os.path.dirname(os.path.abspath(\_\_file\_\_)))**

### **domain\_file = os.path.join(base\_dir, 'domain.pddl')**

### **problem\_files = [**

### **os.path.join(base\_dir, 'problem.pddl'),**

### **os.path.join(base\_dir, 'problem\_jardin.pddl'),**

### **os.path.join(base\_dir, 'problem\_invernadero.pddl'),**

### **os.path.join(base\_dir, 'problem\_huerto.pddl')**

### **]**

### **plans = []**

### **for problem\_file in problem\_files:**

### **# Leer el contenido del archivo problem.pddl**

### **with open(problem\_file, 'r') as file:**

### **problem\_content = file.read()**

### **# Parsear el archivo PDDL seleccionado**

### **parser = Parser(domain\_file, problem\_file)**

### **domain = parser.parse\_domain()**

### **problem = parser.parse\_problem(domain)**

### **# Generar la tarea**

### **task = \_ground(problem)**

### **# Aplicar la heurística y buscar el plan**

### **heuristic = BlindHeuristic(task)**

### **plan = astar\_search(task, heuristic)**

### **# Convertir el plan a una lista de nombres de acciones**

### **plan\_steps = [str(action) for action in plan]**

### **# Extraer el estado inicial del contenido del archivo problem.pddl**

### **init\_start = problem\_content.find('(:init') + len('(:init')**

### **init\_end = problem\_content.find(')', init\_start)**

### **initial\_state\_str = problem\_content[init\_start:init\_end].strip()**

### **# Convertir el objetivo a una cadena de texto legible**

### **goal\_str = "\n".join(sorted(str(goal) for goal in task.goals))**

### **# Agregar detalles del plan**

### **plan\_details = {**

### **'problem\_file': os.path.basename(problem\_file),**

### **'initial\_state': initial\_state\_str,**

### **'goal': goal\_str,**

### **'actions': plan\_steps**

### **}**

### **# Agregar el plan a la lista de planes**

### **plans.append(plan\_details)**

### **# Guardar los planes en la sesión**

### **request.session['plans'] = plans**

### **return redirect('/ver\_plan')**

### **def ver\_plan(*request*):**

### **"""**

### **Muestra los planes generados guardados en la sesión.**

### **"""**

### **plans = request.session.get('plans', [])**

### **return render(request, 'ver\_plan.html', {'plans': plans})**

### urls.py

Este archivo contiene las rutas de URL para la aplicación.

from django.urls import path

from . import views

urlpatterns = [

    path('', views.index, *name*='index'),

    path('generar\_plan', views.generar\_plan, *name*='generar\_plan'),

    path('ver\_plan', views.ver\_plan, *name*='ver\_plan'),

    path('mostrar\_dominio\_y\_problemas', views.mostrar\_dominio\_y\_problemas, *name*='mostrar\_dominio\_y\_problemas'),

    path('heuristica', views.heuristica, *name*='heuristica'),

### problem.pddl

Estos archivos definen los problemas de horticultura en PDDL.

### (define (problem horticultura-problema)

### (:domain horticultura)

### 

### (:objects

### planta1 planta2 planta3 planta4 planta5 - planta

### regadera fertilizante tijeras - herramienta

### jardin invernadero huerto - lugar

### camino1 camino2 camino3 - camino

### )

### 

### (:init

### (en-lugar planta1 jardin)

### (en-lugar planta2 invernadero)

### (en-lugar planta3 huerto)

### (en-lugar planta4 jardin)

### (en-lugar planta5 invernadero)

### (tiene-herramienta regadera)

### (tiene-herramienta fertilizante)

### (tiene-herramienta tijeras)

### (conectado jardin camino1)

### (conectado camino1 invernadero)

### (conectado invernadero camino2)

### (conectado camino2 huerto)

### (conectado huerto camino3)

### (conectado camino3 jardin)

### (en-camino camino1 jardin)

### (en-camino camino1 invernadero)

### (en-camino camino2 invernadero)

### (en-camino camino2 huerto)

### (en-camino camino3 huerto)

### (en-camino camino3 jardin)

### )

### 

### (:goal

### (and

### (cosechada planta1)

### (cosechada planta2)

### (cosechada planta3)

### (cosechada planta4)

### (cosechada planta5)

### )

### )

Problema\_jardin.pddl

(define (problem horticultura-problema-jardin)

(:domain horticultura)

(:objects

planta1 planta4 - planta

regadera fertilizante tijeras - herramienta

jardin - lugar

camino1 - camino

)

(:init

(en-lugar planta1 jardin)

(en-lugar planta4 jardin)

(tiene-herramienta regadera)

(tiene-herramienta fertilizante)

(tiene-herramienta tijeras)

(conectado jardin camino1)

)

(:goal

(and

(cosechada planta1)

(cosechada planta4)

)

)

)

Problema\_huerto.pddl

(define (problem horticultura-problema-huerto)

(:domain horticultura)

(:objects

planta3 - planta

regadera fertilizante tijeras - herramienta

huerto - lugar

camino3 - camino

)

(:init

(en-lugar planta3 huerto)

(tiene-herramienta regadera)

(tiene-herramienta fertilizante)

(tiene-herramienta tijeras)

(conectado huerto camino3)

)

(:goal

(cosechada planta3)

)

)

Problema\_invernadero.pddl

(define (problem horticultura-problema-invernadero)

(:domain horticultura)

(:objects

planta2 planta5 - planta

regadera fertilizante tijeras - herramienta

invernadero - lugar

camino2 - camino

)

(:init

(en-lugar planta2 invernadero)

(en-lugar planta5 invernadero)

(tiene-herramienta regadera)

(tiene-herramienta fertilizante)

(tiene-herramienta tijeras)

(conectado invernadero camino2)

)

(:goal

(and

(cosechada planta2)

(cosechada planta5)

)

)

)

### )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  
 )  
)

HTMLs  
  
dominio\_y\_problemas.html  
  
muestra los contenidos del dominio y los problemas en una pagina

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <title>Dominio y Problemas</title>

    <style>

        body {

            font-family: Arial, sans-serif;

            margin: 20px;

        }

        h1, h2 {

            color: #333;

        }

        pre {

            background: #f9f9f9;

            padding: 10px;

            border: 1px solid #ddd;

            overflow: auto;

        }

        .problem-container {

            margin-bottom: 20px;

        }

    </style>

</head>

<body>

    <h1>Dominio y Problemas</h1>

    <h2>Dominio</h2>

    <pre>{{ domain\_content }}</pre>

    <h2>Problemas</h2>

    {% for problem\_type, problem\_content in problems.items %}

        <div class="problem-container">

            <h3>{{ problem\_type|capfirst }}</h3>

            <pre>{{ problem\_content }}</pre>

        </div>

    {% endfor %}

</body>

</html>

Heuristica.html

Muestra el análisis de la heurística definida para los 3 problemas  
  
<!DOCTYPE html>

<html>

<head>

    <title>Heurística Aplicada</title>

    <link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/bootstrap/4.3.1/css/bootstrap.min.css">

</head>

<body>

    <div class="container">

        <h1>Heurística Aplicada</h1>

        <pre>{{ heuristica\_content }}</pre>

        <a href="/" class="btn btn-primary mt-3">Volver</a>

    </div>

</body>

</html>

Index.html

Pagina principal

<!DOCTYPE html>

<html>

<head>

    <title>Planificador de Horticultura</title>

    <link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/bootstrap/4.3.1/css/bootstrap.min.css">

</head>

<body>

    <div class="container">

        <h1>Planificador de Horticultura</h1>

        <form method="post">

            {% csrf\_token %}

            <button type="submit" name="action" value="generar\_plan" class="btn btn-primary">Generar Plan</button>

            <a href="{% url 'ver\_plan' %}" class="btn btn-secondary">Ver Plan</a>

            <a href="{% url 'heuristica' %}" class="btn btn-warning">Aplicar Heurística</a>

            <a href="{% url 'mostrar\_dominio\_y\_problemas' %}" class="btn btn-info">Mostrar Dominio y Problemas</a>

        </form>

    </div>

</body>

</html>

ver\_plan.html

vista que muestra el plan generado por cada problema  
  
<!DOCTYPE html>

<html lang="es">

<head>

    <meta charset="UTF-8">

    <title>Planes Generados</title>

    <style>

        body {

            font-family: Arial, sans-serif;

            background-color: #f4f4f4;

            margin: 0;

            padding: 0;

        }

        .container {

            width: 80%;

            margin: 0 auto;

            padding: 20px;

            background-color: #fff;

            box-shadow: 0 0 10px rgba(0, 0, 0, 0.1);

            border-radius: 8px;

            margin-top: 20px;

        }

        h1 {

            text-align: center;

            color: #333;

        }

        .plan {

            margin-bottom: 20px;

            padding: 15px;

            border: 1px solid #ddd;

            border-radius: 8px;

            background-color: #fafafa;

        }

        .plan h2 {

            color: #555;

        }

        .plan p {

            margin: 5px 0;

        }

        .plan ul {

            list-style-type: none;

            padding: 0;

        }

        .plan ul li {

            background-color: #e7f3fe;

            margin: 5px 0;

            padding: 10px;

            border-radius: 4px;

        }

        .state, .goal {

            background-color: #f9f9f9;

            padding: 10px;

            border-radius: 4px;

            margin-bottom: 10px;

        }

        .state p, .goal p {

            margin: 0;

        }

    </style>

</head>

<body>

    <div class="container">

        <h1>Planes Generados</h1>

        {% for plan in plans %}

            <div class="plan">

                <h2>Plan para {{ plan.problem\_file }}</h2>

                <div class="state">

                    <p><strong>Estado Inicial:</strong></p>

                    <pre>{{ plan.initial\_state }}</pre>

                </div>

                <div class="goal">

                    <p><strong>Objetivo:</strong></p>

                    <pre>{{ plan.goal }}</pre>

                </div>

                <p><strong>Acciones:</strong></p>

                <ul>

                    {% for action in plan.actions %}

                        <li>{{ action }}</li>

                    {% endfor %}

                </ul>

            </div>

        {% endfor %}

        <a href="/" class="btn btn-primary mt-3">Volver</a>

    </div>

</body>

</html>

## 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.