

|  |  |
| --- | --- |
| Tercera Pre Entrega | |
|  |  |
| 31-Julio-2023Edwin Vargas |  |

|  |  |  |
| --- | --- | --- |
|  |  |  |
| INSTRUCCIÓN | | |
| En esta pre entrega trataré de explicar paso a paso como crear un proyecto Web en Django, HTML y CSS  Crearé un entorno virtual (env) para almacenar las versiones que descargue, adicionalmente a Django, instalo Pillow para el manejo de las imágenes.  El proyecto se trata de una Agencia de Viajes que maneja como productos paquetes turisticos.  Nuestro proyecto principal se llamará core con una APP con el nombre de categories  Se podrá crear categorías y elaboraremos un CRUD para los productos.  Deberé crear un archivo form.py para crear categorías y en productos crear y editar.  Para la creación de los archivos HTML y CSS me apoyare en Bootstrap | | |  |
|  | | |

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | |  |  |  |  | | --- | --- | --- | --- | |  | **✨** **Instalamos:**   * **Python:** <https://www.python.org/> * **Git:** https://git-scm.com/ * **Github:** <https://github.com/>   **✨ Creamos carpeta para el Proyecto.**  Creamos una carpeta con el nombre del Proyecto. Dentro del terminal escribimos el siguiente código.  Mkdir WEB\_AV  cd WEB\_AV  **✨ Creamos entorno virtual.**  Para realizar un seguimiento de todos los paquetes que instalamos durante el desarrollo de este proyecto.  ……\WEB\_AV>  python -m virtualenv env  Estaré trabajando en Windows por ende utilizaré los comandos de esta terminal.  ……\WEB\_AV>  env/scripts/activate  **✨ Instalamos Django y Pillow**  Pillow es una biblioteca que nos permite abrir, manipular y guardar muchos formatos de archivo de imagen diferentes.  (env)……\WEB\_AV>  pip install Django  pip install Django Pillow  Creamos un archivo ***requirements.txt*** con los siguientes comandos.  (env)……\WEB\_AV>  pip freeze  pip freeze > requirements.txt  Se crea el archivo ***requirements.txt*** con los siguientes paquetes.  WEB\_AV\requirements.txt  asgiref==3.7.2  Django==4.2.3  Pillow==10.0.0  sqlparse==0.4.4  tzdata==2023.3  **✨ Creamos el Proyecto.**  El nombre del proyecto será **core**.  (env)……\WEB\_AV>  django-admin startproject core .  Este código genera automáticamente, la estructura de carpetas/ficheros como se muestra abajo:    Dentro de la carpeta principal **WEB\_AV** creamos dos carpetas ***apps*** para crear aquínuestras apps y ***static*** para nuestros archivos ***html, js e imágenes.***  (env)……\WEB\_AV>  mkdir apps  mkdir static  Dentro de la carpeta **apps** creamos un archivo ***\_\_init\_\_.py***  Dentro de la carpeta **static** creamos las carpetas: css, img y js    Creamos la estructura para las diferentes apps en el archivo ***settings.py.***  (env)……\WEB\_AV\core\setting.py  # Application definition  DJANGO\_APPS = [      'django.contrib.admin',      'django.contrib.auth',      'django.contrib.contenttypes',      'django.contrib.sessions',      'django.contrib.messages',      'django.contrib.staticfiles',  ]  PROJECT\_APPS = [      'apps.categories',  ]  THIRD\_PARTY\_APPS = [  ]  INSTALLED\_APPS = DJANGO\_APPS + PROJECT\_APPS + THIRD\_PARTY\_APPS  ]  INSTALLED\_APPS = DJANGO\_APPS + PROJECT\_APPS + THIRD\_PARTY\_APPS    **✨ Creamos nuestra primera app** ***categories.***  Debido a que las aplicaciones no estarán en el mismo directorio manage.py, añadimos al código (..\)  ….cd apps    (env)……\WEB\_AV\apps>  Python ..\manage.py startapp categories    Esta es la estructura de archivos que se crea.    Modificamos el archivo ***apps.py*** de nuestras apps. (name = 'apps. categories’)  (env)……\WEB\_AV\apps\categories\apps.py  from django.apps import AppConfig  class CategoriesConfig(AppConfig):      default\_auto\_field = 'django.db.models.BigAutoField'      name = 'apps.categories'  Agregamos las siguientes funciones en el archivo ***views.py.*** Con la finalidad de hacer nuestras primeras pruebas.  (env)……\WEB\_AV\apps\categories\views.py  from django.shortcuts import render  from django.http import HttpResponse  def home(request):      return HttpResponse('Inicio')  def about (request):      return HttpResponse('Quienes Somos!')  def category\_list (request):      return HttpResponse('Categoria Listar')  def category\_create (request):      return HttpResponse('Categoria Crear')  def product\_list(request):      return HttpResponse('Producto Listar')  def product\_read(request):      return HttpResponse('Producto Detalle')  def product\_create(request):      return HttpResponse('Producto Crear')  def product\_update(request):      return HttpResponse('Producto Editar')  def product\_delete(request):      return HttpResponse('Producto Eliminar')  **✨ Creamos archivo *urls.py* en categories.**  Para re direccionar las funciones de nuestra vista creamos la estructura.  (env)……\WEB\_AV\apps\categories\urls.py  from django.urls import path  from . import views  from django.urls import path  from . import views  urlpatterns = [      path('', views.home, name='home'),      path('about/', views.about,name = 'about' ),      path('category\_list/', views.category\_list, name = 'category\_list'),      path('category\_create/', views.category\_create, name='category\_create'),      path('product\_list/', views.product\_list, name = 'product\_list'),      path('product\_create/', views.product\_create, name='product\_create'),      path('product\_read/<id>/', views.product\_read, name='product\_read'),      path('product\_update/<id>/', views.product\_update, name='product\_update'),      path('product\_delete/<id>/', views.product\_delete, name='product\_delete'),  ]  Dentro del archivo ***urls.py*** que está en el proyecto core,direccionamos las urls de nuestra app categories.  (env)……\WEB\_AV\core\urls.py  from django.contrib import admin  from django.urls import path, include  urlpatterns = [      path('admin/', admin.site.urls),      path('categories/', include('apps.categories.urls')),  ]  **✨ Hacemos las pruebas de las urls que creamos.**  (env)……\ WEB\_AV>  python manage.py runserver  http://127.0.0.1:8000/    url: http://127.0.0.1:8000/categories    url: http://127.0.0.1:8000/categories/about/    url: http://127.0.0.1:8000/categories/product\_list/    **✨ Configuramos los archivos estáticos:**  Dentro del archive ***setting.py*** de nuestro proyecto core,debajo de STATIC\_URLagregamos el siguiente código.  (env)……\WEB\_AV\core\setting.py  STATIC\_URL = 'static/'  MEDIA\_URL = '/img/'  MEDIA\_ROOT = BASE\_DIR / 'static/'  STATICFILES\_DIRS = [ BASE\_DIR / 'static' ]  **✨ Creamos los modelos.**  Crearemos dos clases Category: Con el nombre del País y Poduct: donde estarán cada uno de los productos que desarrollemos, incluye un FK que lo asocia a Category.  En nuestra app categories en el archivo ***models.py*** agregamos nuestras clases.  (env)……\WEB\_AV\apps\destinations\models.py  from django.db import models  class Category(models.Model):      name = models.CharField(max\_length=150)      class Meta:          ordering = ['name']          verbose\_name = ("Category")          verbose\_name\_plural = ("Categories")      def \_\_str\_\_(self):          return self.name    class Product(models.Model):      title = models.CharField(max\_length=150)      content = models.TextField(max\_length=2000, blank=True)      category = models.ForeignKey(Category, on\_delete=models.CASCADE)      image = models.ImageField(blank = True, upload\_to = 'img/')        class Meta:          ordering = ["-title"]      def \_\_str\_\_(self):          return self.title  **✨ Generamos las migraciones**  Ahora ejecutamos este comando para generar un script de migraciones para la aplicación categories. Esto generará el script 0001\_initial.py dentro de las carpetas de las apps /migrations para todos los modelos que he definido en /models.py. No se debe realizar cambios en este script.  (env)……\WEB\_AV>  python manage.py makemigrations  python manage.py migrate  Nos debe arrojar el siguiente resultado por pantalla.    **✨ Creamos ‘superuser’ en Administrador de Django**  Creamos un super usuario con el siguiente comando:  (env)……\WEB\_AV>  python manage.py createsuperuser  Username: admin  Email address: admin@gmail.com  Password: admin123  Modificamos el archivo ***admin.py*** en la app **categories.** La documentación la podemos ver en este enlace: https://docs.djangoproject.com/en/4.2/ref/contrib/admin/  (env)……\WEB\_AV\apps\categories\admin.py  from django.contrib import admin  from .models import Category, Product  admin.site.register(Category)  admin.site.register(Product)  Iniciar sesión en nuestro administrador de Django.    Dentro de categories agregamos 6 Países:    **✨ Creamos archivos .html**  Creamos una carpeta **templates** dentro de apps. Dentro de **templates** creamos otra carpeta con el nombre de la app **categories** y agregamos los siguientes archivos html.   * base.html * home.html * about.html * category\_list.html * category\_create.html * product\_list.html * product\_create.html * product\_read.html * product\_update.html * product\_delete.html   Direccionamos nuestros archivos de la carpeta templates.  (env)……\WEB\_AV\core\setting.py  TEMPLATES = [      {          '……',          'DIRS': [BASE\_DIR/'templates'],  '……',  Agregamos el siguiente código al archivo ***base.html.***  (env)……\DBVE\apps\templates\categories\base.html  <!DOCTYPE html>  <html lang="en">    <head>      <meta charset="utf-8">      <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no">      <link href="https://fonts.googleapis.com/css?family=Roboto:300,400&display=swap" rel="stylesheet">      <link href="https://fonts.googleapis.com/css?family=Poppins:300,400,500&display=swap" rel="stylesheet">      <link href="https://fonts.googleapis.com/css?family=Source+Serif+Pro:400,600&display=swap" rel="stylesheet">      <link rel="stylesheet" href="/static/css/style.css">      <link rel="stylesheet" href="/static/css/bootstrap.min.css">      <title>Agencia AV</title>    </head>    <body>  <!-- Navbar -->  <section>    <nav class="navbar navbar-expand-lg bg-dark navbar-dark ">       <div class="container-fluid justify-content-center">        <a class="navbar-brand" href="{% url 'home' %}"> Agencia AV</a>          <div class="collapse navbar-collapse" id="navbarResponsive">            <ul class="navbar-nav mr-auto mt-2 mt-lg-0">              <li class="nav-item"><a class="nav-link" href="{% url 'product\_list' %}">Destinos</a></li>              <li class="nav-item"><a class="nav-link" href="{% url 'about' %}">Nosotros</a></li>              <li class="nav-item"><a class="nav-link" href="{% url 'admin:index' %}">Adminin</a>            </ul>            <ul class="nav navbar-nav my-2 my-lg-0">              <li class="nav-item"><a class="nav-link" href="#"><i class="fas fa-sign-in-alt"></i> Login</a></li>              <li class="nav-item"><a class="nav-link" href="#"><i class="fas fa-sign-in-alt"></i> Logout</a></li>            </ul>            <i class="mobile-nav-toggle d-xl-none bi bi-list"></i>          </div>      </div>    </nav>  </section>    <body>        {% block content %}          {% endblock %}    </body>    <!-- footer Pendiente -->    </body>  </html>  Agregamos el siguiente código al archivo [***home.html.*** La página está en construcción solo tiene una linda foto del Yaque en Venezuela.](http://home.html.Por)  (env)……\DBVE\apps\templates\categories\home.html  {% extends 'categories/base.html' %}  {% load static %}  {% block content %}  <html>  <body>      <div class="has-bg-img bg-secondary bg-blend-screen">          <div class="bg-img bg-cover">              <img src="/static/img/20220113\_180308.JPG" class="img-fluid img-thumbnail" alt="...">              <h4>Estas en Home: En Construccion</h4>          </div>      </div>  </body>  </html>  {% endblock %}  Agregamos el siguiente código al archivo ***about.html.***  (env)……\DBVE\apps\templates\categories\about.html  {% extends 'categories/base.html' %}  {% load static %}  {% block content %}      <div class="site-mobile-menu site-navbar-target">        <div class="site-mobile-menu-header">          <div class="site-mobile-menu-close">            <span class="icofont-close js-menu-toggle"></span>          </div>        </div>        <div class="site-mobile-menu-body"></div>      </div>      <div class="section">        <div class="container">          <div class="row text-left mb-5">            <div class="col-12">              <h2 class="font-weight-bold heading text-primary mb-6">Quienes Somos</h2>            </div>            <div class="col-lg-6">              <p class="text-black-50">                Aunque la exploración a menudo se asocia con lugares, en el fondo,                se trata de las personas. Durante 20 años, la experiencia de persona                a persona ha sido el núcleo de la misión de AV. Es lo que siempre                hemos hecho y es lo que hacemos ahora. Gente real ayudando a gente real.                A tu lado en cada paso del camino.              </p>            </div>          </div>        </div>      </div>  {% endblock %}  Agregamos el siguiente código al archivo ***category\_list.html.***  (env)……\DBVE\apps\templates\categories\category\_list.html  {% extends 'categories/base.html' %}  {% load static %}  {% block content %}  <div class="container mb-4">      <div class="row">          <div class="col-12">              <h2 class="font-weight-bold heading text-primary mb-6">Pais Destino</h2>          </div>          <div class="col-md-3 mt-3">              <div class="card">                  <a href="{% url 'category\_create' %}" class="m-2 btn btn-dark btn-sm">Agregar Pais</a>              </div>          </div>          <div class="col-md-9 mt-3">              <div class="row">                  {% for categories in categories %}                  <div class="col-md-4">                          <h4 class="card-text">{{ categories.name }}</h4>                  </div>                  {% endfor %}              </div>          </div>      </div>  </div>  {% endblock %}  Agregamos el siguiente código al archivo ***category\_create.html.***  (env)……\DBVE\apps\templates\categories\about.html  {% extends 'categories/base.html' %}  {% block content %}  <div class="container mb-6">      <div class="row">          <div class="col-12">              <h2 class="font-weight-bold heading text-primary mb-6">Agregar Pais</h2>          </div>          <div class="col-md-3 mt-3">              <div class="card">                  <a href="{% url 'category\_list' %}" class="m-2 btn btn-dark btn-sm">Regresar</a>              </div>          </div>          <div class="col-md-4">              <div class="card shadow-lg">                  <div class="card-body">                      <form method="POST" action="{% url 'category\_create' %}" enctype="multipart/form-data">                          {% csrf\_token %}                          <p>{{ form }}</p>                          {% for error in field.errors %}                          <p class="alarma">{{error}}</p>                          {% endfor %}                          {% for error in form.non\_field\_errors %}                          <p class="alarma">{{error}}</p>                          {% endfor %}                          <div class="modal-footer bg-white">                              <button type="submit" class="btn btn-success">Agregar</button>                          </div>                      </form>                  </div>              </div>          </div>      </div>  </div>  {% endblock %}  Agregamos el siguiente código al archivo ***product\_list.html.***  (env)……\DBVE\apps\templates\categories\product\_list.html  {% extends 'categories/base.html' %}  {% load static %}  {% block content %}  <div class="container mb-4">      <div class="row">          <div class="col-12">              <h2 class="font-weight-bold heading text-primary mb-6">Destinos</h2>          </div>          <div class="col-md-3 mt-3">              <div class="card">                  <a href="{% url 'product\_create' %}" class="m-2 btn btn-dark btn-sm">Agregar Destino</a>              </div>          </div>          <div class="col-md-9 mt-3">              <div class="row">                    {% for products in products %}                  <div class="col-md-4">                      <div class="card my-2">                          <a href=""><img src="{{ products.image.url }}" class="card-img-top" style="height:275px" /></a>                      <div class="card-body">                          <h4 class="card-text">{{ products.title }}</h4>                      </div>                          <a href="{% url 'product\_read' products.id %}" class="m-2 btn btn-outline-dark btn-sm">Ver Mas</a>                      </div>                  </div>                  {% endfor %}              </div>          </div>      </div>  </div>  {% endblock %}  Agregamos el siguiente código al archivo ***producto\_create.html.***  (env)……\DBVE\apps\templates\categories\product\_create.html  {% extends 'categories/base.html' %}  {% block content %}  <div class="container mb-6">      <div class="row">          <div class="col-12">              <h2 class="font-weight-bold heading text-primary mb-6">Agregar Destino</h2>          </div>          <div class="col-md-3 mt-3">              <div class="card">                  <a href="{% url 'product\_list' %}" class="m-2 btn btn-dark btn-sm">Regresar</a>              </div>          </div>          <div class="col-md-4">              <div class="card shadow-lg">                  <div class="card-body">                      <form method="POST" action="{% url 'product\_create' %}" enctype="multipart/form-data">                          {% csrf\_token %}                          <p>{{ form }}</p>                          {% for error in field.errors %}                          <p class="alarma">{{error}}</p>                          {% endfor %}                          {% for error in form.non\_field\_errors %}                          <p class="alarma">{{error}}</p>                          {% endfor %}                          <div class="modal-footer bg-white">                              <button type="submit" class="btn btn-success">Agregar</button>                          </div>                      </form>                  </div>              </div>          </div>      </div>  </div>  {% endblock %}  Agregamos el siguiente código al archivo ***producto\_read.html.***  (env)……\DBVE\apps\templates\categories\product\_read.html  { {% extends 'categories/base.html' %}  {% load static %}  {% block content %}  <div class="container mb-4">          <div class="row">              <div class="col-12">                  <h2 class="font-weight-bold heading text-primary mb-6">Detalle Destinos</h2>              </div>              <div class="col-md-3 mt-3">                  <div class="card">                      <a href="{% url 'product\_list' %}" class="m-2 btn btn-dark btn-sm">Regresar</a>                      <a href="{% url 'product\_update' products.id %}" class="m-2 btn btn-success btn-sm">Editar Destino</a>                      <a href="{% url 'product\_delete' products.id %}" class="m-2 btn btn-danger btn-sm">Eliminar Destino</a>              </div>          </div>              <div class="col-md-4">                  <div style="height: 90vh">                      <img class="img-fluid text-center" style="max-width: 100%; max-height: 100%" src="{{ products.image.url }}" alt="">                      <h5 class="text-muted">Destino: {{ products.title }}</h5>                      <small>Contenido: {{ products.content }}</small><br>                  </div>              </div>          </div>      </div>  {% endblock %}  Agregamos el siguiente código al archivo ***product\_update.html.***  (env)……\DBVE\apps\templates\categories\product\_update.html  {% extends 'categories/base.html' %}  {% block content %}  <div class="container mb-6">      <div class="row">          <div class="col-12">              <h2 class="font-weight-bold heading text-primary mb-6">Editar Destino</h2>          </div>          <div class="col-md-3 mt-3">              <div class="card">                  <a href="{% url 'product\_list' %}" class="m-2 btn btn-dark btn-sm">Regresar</a>              </div>          </div>          <div class="col-md-4">              <div class="card shadow-lg">                  <div class="card-body">                      <form method="POST" enctype="multipart/form-data" action="">                          {% csrf\_token %}                          <p>{{ form }}</p>                          <button type="submit" class="btn btn-success">Editar Producto</button>                      </form>                  </div>              </div>          </div>      </div>  </div>  {% endblock %}  Agregamos el siguiente código al archivo ***product\_delete.html.***  (env)……\DBVE\apps\templates\categories\product\_delete.html  {% extends 'categories/base.html' %}  {% block content %}  <div class="container mb-6">      <div class="row">          <div class="col-12">              <h2 class="font-weight-bold heading text-primary mb-6">Eliminar Destino</h2>          </div>          <div class="col-md-3 mt-3">              <div class="card">                  <a href="{% url 'product\_list' %}" class="m-2 btn btn-dark btn-sm">Regresar</a>              </div>          </div>          <div class="col-md-4">              <div class="card shadow-lg">                  <div class="card-body">                      <form method="POST" action="">                          {% csrf\_token %}                          <p>Quieres eliminar este destino? '<b>{{ products.title | upper}}</b>'?</p>                          <button type="submit" class="btn btn-danger btn-block mb-4">Confirmar</button>                      </form>                  </div>              </div>          </div>      </div>  </div>  {% endblock %}  **✨ Creamos archivo forms.**  Dentro de nuestra aplicación categories creamos un archivo ***forms.py*** y agregamos las siguientes clases. Estos serán nuestros formularios.  (env)……\WEB\_AV\apps\categories\forms.py  from .models import Category, Product  from django.forms import ModelForm  from django import forms  class CategoryForm(forms.ModelForm):      class Meta:          model = Category          fields = ('name', )  class CreateCategoryForm(forms.ModelForm):      class Meta:          model = Category          fields = ('name',)          labels = {              'name': 'Pais Destino ',          }  class CreateProductForm(forms.ModelForm):      class Meta:          model = Product          fields = ('title', 'content', 'category', 'image')          labels = {              'title': 'Destino ',              'content': 'Detalle Destino ',              'category': 'Pais ',              'image': 'Imagen ',          }  class UpdateProductForm(ModelForm):      class Meta:          model = Product          fields =  ('title', 'content', 'category', 'image')          labels = {              'title': 'Destino ',              'content': 'Detalle Destino ',              'category': 'Pais ',              'image': 'Imagen ',          }  Realizamos los siguientes cambios en ***views.py*** de nuestra app categories.  (env)……\WEB\_AV\apps\categories\views.py  from django.shortcuts import render, redirect  from .models import Category, Product  from .forms import \*  from django.contrib import messages  def home(request):      return render(request, 'categories/home.html')  def about (request):      return render(request, 'categories/about.html')  def category\_list (request):      categories = Category.objects.order\_by('name')      context = {          'categories': categories,      }      return render(request, 'categories/category\_list.html', context)  def category\_create(request):      if request.method == 'POST':          form = CreateCategoryForm(request.POST, request.FILES )          if form.is\_valid:              try:                  form.save()                  return redirect('category\_list')              except:                  messages(request, 'Hay un error en los datos suministrados')      else:          form = CreateCategoryForm()      return render(request, 'categories/category\_create.html', {"form":form})  def product\_list(request):      products = Product.objects.order\_by('-title')      context = {          'products': products,      }      return render(request, 'categories/product\_list.html', context)  def product\_read(request, id):      products = Product.objects.get(pk = id)      context = {'products': products}      return render(request, 'categories/product\_read.html', context)  def product\_create(request):      if request.method == 'POST':          form = CreateProductForm(request.POST, request.FILES )          if form.is\_valid:              try:                  form.save()                  return redirect('product\_list')              except:                  messages(request, 'Hay un error en los datos suministrados')      else:          form = CreateProductForm()      return render(request, 'categories/product\_create.html', {"form":form})  def product\_update(request, id):      products = Product.objects.get(pk = id)      form = UpdateProductForm(instance = products)      if request.method == 'POST':          form = UpdateProductForm(request.POST, request.FILES, instance = products)          if form.is\_valid():              form.save()              return redirect('product\_list')      context = {'form': form}      return render(request, 'categories/product\_update.html', context)  def product\_delete(request, id):      products = Product.objects.get(pk = id)      if request.method == 'POST':          products.delete()          return redirect('home')      context = {'products': products}      return render(request, 'categories/product\_delete.html', context)  Dentro del archivo ***urls.py*** que está en el proyecto core,debemos hacer los siguientes cambios ya que estaremos trabajando con imágenes que están dentro de la carpeta static.  Para ver la documentación: https://docs.djangoproject.com/en/4.2/howto/static-files/  (env)……\WEB\_AV\core\urls.py  from django.contrib import admin  from django.urls import path, include  from django.conf import settings  from django.conf.urls.static import static  urlpatterns = [      path('admin/', admin.site.urls),      path('categories/', include('apps.categories.urls')),  ]  urlpatterns += static(settings.MEDIA\_URL, document\_root = settings.MEDIA\_ROOT)  **✨ Agregamos los estilos e imágenes en nuestra carpeta css*.***  **css:**  bootstrap.min.css: <https://getbootstrap.com/docs/5.0/getting-started/download/>  style.css: https: <https://www.cdnpkg.com/linearicons/file/style.css/>  **✨ Creamos 2 Productos.**  **Destino:** Kite y Windsurf El Yaque Isla de Margarita  **País:** Venezuela  **Detalle Destino:**   * Boletos aéreos hacia Isla de Margarita desde Maiquetía (Caracas) con retorno. * Traslado al Yaque y Retorno * Alojamiento en posada. * Club de Windsurf y Kite Surf * Sillas y Toldos de Playa * Desayunos y cenas en la posada.   **Imagen:** 20220113\_180308.JPG  **Producto:** Buceo en El Archipiélago Los Testigos.  **Destino:** Venezuela  **Descripsion:**   * Traslado marítimo desde el Tirano Isla de Margarita al Archipiélago de los Testigos y regreso en embarcaciones tipo peñero. * Inmersiones diurnas y nocturnas con instructor (2 inmersiones diarias) * Alojamiento matrimonial o individual en Posada de Los Testigos. * Tanques, bote para buceo, guía, refrescos, desayunos, almuerzos y cenas. En Los Testigos. * Por medidas de seguridad no se permiten bebidas alcohólicas.   **Imagen:** SAM\_0994.JPG  Ejemplo del Formulario que creamos para la carga de Destinos.    Lista de destinos:    Detalle Destinos:    Editar Destinos:    Eliminar destino: | |  | |  |  |  | | |  |