

ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL

Curso DESARROLLO DE APLICACIONES WEB Y MÓVILES - II PAO

2025

Evaluación: Examen Segundo Parcial

Codigo de Honor - (0.0 punto(s))

Yo, en mi calidad de estudiante de la ESPOL, declaro que he sido informado y conozco las normas que rigen a la ESPOL, en particular el Código de Ética y el Reglamento de Disciplina.

Acepto con pleno conocimiento este compromiso de honor por el cual reconozco y estoy consciente de que la presente evaluación está diseñada para ser resuelta de forma individual; que sólo puedo comunicarme con la persona responsable de la recepción de la evaluación; que no haré consultas indebidas o no autorizadas por el evaluador; ni usaré dispositivos electrónicos.

Acepto el presente compromiso, como constancia de haber leído y aceptar la declaración anterior y me comprometo a seguir fielmente las instrucciones que se indican para la realización de la presente evaluación.

Recuerde:

- Ud. fue informado y acepto el código de honor.
- Ud. fue notificado sobre la importancia de grabar la pantalla (en su totalidad) mientras realiza el examen.
- La evaluación es de carácter individual y se comprometió a:
 - Realizarla sin navegar en sitios web fuera del Aula Virtual.
 - Solo utilizar las aplicaciones/Software autorizados para esta evaluación
 - No solicitar ni recibir asistencia externa, ya sea de otras personas o de motores de inteligencia artificial, presencial o virtual.
 - Abstenerse de consultar libros, notas, o cualquier otro material.
 - No emplear dispositivos electrónicos o de comunicación.

Al proceder con esta evaluación, usted afirma haber leído, entendido y aceptado los términos de esta declaración, asimismo, reconoce que el incumplimiento del presente compromiso, anulará automáticamente

Identificar código problemático en Drawer - (1.0 punto(s))

¿Cuál línea provoca un problema en la navegación y por qué?

```
<Drawer.Navigator initialRouteName="Inicio">  
  <Drawer.Screen name="Inicio" component={HomeScreen} />  
  <Drawer.Screen name="Inicio" component={ProfileScreen} />  
</Drawer.Navigator>
```

DETECCIÓN Y CORRECCIÓN DE ERROR EN TAB - (1.0 punto(s))

Se está desarrollando una aplicación móvil en React Native utilizando React Navigation con navegación por Tabs.

El requerimiento funcional indica que la aplicación debe iniciar en la pestaña Lista.

Se le entrega el siguiente código, el cual presenta un error de configuración que impide cumplir el requerimiento:

```
import React from "react";
import { NavigationContainer } from "@react-navigation/native";
import { createBottomTabNavigator } from "@react-navigation/bottom-tabs";
import ListScreen from "../screens/ListScreen";
import SettingsScreen from "../screens/SettingsScreen";
const Tab = createBottomTabNavigator();
export default function App() {
  return (
    <NavigationContainer>
      <Tab.Navigator initialRouteName="Listado">
        <Tab.Screen
          name="Lista"
          component={ListScreen}
        />
        <Tab.Screen
          name="Configuracion"
          component={SettingsScreen}
        />
      </Tab.Navigator>
    </NavigationContainer>
  );
}
```

El Tab Navigator no iniciará en la pestaña esperada.

1. Identifique el error presente en la configuración del Tab Navigator.
2. Escriba la corrección mínima necesaria para que la aplicación inicie correctamente en la pestaña Lista.

Nota:

- No agregue nuevas pantallas.
- No modifique componentes ni imports.
- Realice únicamente la corrección necesaria.

TAB (NIVEL BÁSICO / GUIADO) - (1.0 punto(s))

Se está desarrollando una aplicación móvil en React Native utilizando React Navigation.

El requerimiento funcional indica que la aplicación debe usar navegación por pestañas (Tab) para permitir al usuario cambiar entre dos pantallas principales.

Código base (incompleto):

```
import React from "react";
import { NavigationContainer } from "@react-navigation/native";
import { createBottomTabNavigator } from "@react-navigation/bottom-tabs";
import ListScreen from "../screens/ListScreen";
import SettingsScreen from "../screens/SettingsScreen";
const Tab = createBottomTabNavigator();
export default function App() {
  return (
    <NavigationContainer>
      <Tab.Navigator initialRouteName="Listado">
        <Tab.Navigator>
          <Tab.Screen
            name="Lista"
            component={ListScreen}
          />
          <Tab.Screen
            name="Configuracion"
            component={SettingsScreen}
          />
        </Tab.Navigator>
      </Tab.Navigator>
    </NavigationContainer>
  );
}
```

Complete manualmente el código para que la navegación por Tabs cumpla con los siguientes requisitos:

1. Deben existir dos pestañas llamadas Listado y Configuración.
2. La pestaña Listado debe cargar el componente ListScreen.
3. La pestaña Configuración debe cargar el componente SettingsScreen.

Nota:

- No agregue estilos, iconos ni pantallas adicionales.
- No modifique ninguna otra parte del código base.

DETECCIÓN Y CORRECCIÓN DE ERROR EN DRAWER - (1.0 punto(s))

Se está desarrollando una aplicación móvil en React Native utilizando React Navigation con navegación por Drawer.

El requerimiento funcional indica que la aplicación debe iniciar en la pantalla Home al ejecutarse.

Se le entrega el siguiente código, el cual presenta una inconsistencia de configuración que impide cumplir el requerimiento:

```
import React from "react";
import { NavigationContainer } from "@react-navigation/native";
import { createDrawerNavigator } from "@react-navigation/drawer";
import HomeScreen from "../screens/HomeScreen";
import ProfileScreen from "../screens/ProfileScreen";
const Drawer = createDrawerNavigator();

export default function App() {
  return (
    <NavigationContainer>
      <Drawer.Navigator initialRouteName="Inicio">
        <Drawer.Screen
          name="Home"
          component={HomeScreen}
        />
        <Drawer.Screen
          name="Perfil"
          component={ProfileScreen}
        />
      </Drawer.Navigator>
    </NavigationContainer>
  );
}
```

Enunciado

El código anterior no inicia en la pantalla esperada.

1. Identifique el error presente en la configuración del Drawer Navigator.
2. Escriba la corrección mínima necesaria para que la aplicación inicie correctamente en la pantalla Home.

Nota:

- No reescriba todo el archivo.
- No agregue nuevas pantallas ni modifique componentes.
- Realice únicamente la corrección necesaria.

Configuración problemática en Drawer - (1.0 punto(s))

Se está desarrollando una aplicación móvil en React Native que utiliza navegación por Drawer. El requerimiento indica que el menú lateral debe permitir acceder correctamente a todas las pantallas y que la aplicación debe iniciar en la opción Inicio.

El código no cumple correctamente el requerimiento.

```
<Drawer.Navigator initialRouteName="Inicio">
  <Drawer.Screen name="Inicio" component={HomeScreen} />
  <Drawer.Screen name="Perfil" component={ProfileScreen} />
  <Drawer.Screen name="Inicio" component={SettingsScreen} />
  <Drawer.Screen name="Ayuda" component={HelpScreen} />
</Drawer.Navigator>
```

¿Cuál es el problema principal y por qué?

Orden correcto de navegación con Drawer - (1.0 punto(s))

Se está desarrollando una aplicación móvil en React Native utilizando React Navigation con Drawer. El requerimiento indica que la aplicación debe iniciar en la pantalla Inicio y que el menú lateral debe funcionar correctamente.

¿Cuál es el orden correcto de los fragmentos para que la navegación con Drawer funcione correctamente?

- A) <Drawer.Screen name="Inicio" component={HomeScreen} />
- B) <NavigationContainer>
- C) </Drawer.Navigator>
- D) <Drawer.Navigator initialRouteName="Inicio">
- E) </NavigationContainer>

Ventaja técnica de Django - (1.0 punto(s))

¿Cuál es una ventaja técnica real de utilizar Django en un proyecto web?

Vista como centro del flujo - (1.0 punto(s))

En Django, una vista se considera el punto central del flujo de una petición porque:

Rol conceptual de la Vista - (1.0 punto(s))

Complete conceptualmente la frase correcta sobre el rol de una vista:

“Una vista en Django no define diseño visual porque su responsabilidad principal es _____.”

Evaluación de respuesta HTTP - (1.0 punto(s))

Complete correctamente el código para evaluar si una respuesta HTTP fue exitosa:

```
fetch(url)
  .then(response => {
    if (response.____) {
      return response.json();
    }
  });
```


Petición HTTP con fetch - (1.0 punto(s))

Analice el siguiente fragmento JavaScript:

```
fetch("/api/productos", {  
  method: "POST",  
  headers: {  
    "Content-Type": "application/json"  
  },  
  body: JSON.stringify({ nombre: "Mouse", precio: 10 })  
});
```

¿Cuál afirmación es técnicamente correcta respecto a esta petición HTTP?

Idea central del enrutamiento - (1.0 punto(s))

Complete correctamente la idea central del enrutamiento en Django:

“Si una URL no está declarada en `urls.py`, entonces _____.”

Separación de URLs por app - (1.0 punto(s))

¿Cuál es la razón técnica principal para separar las URLs por app en Django?

Asociación URL y Vista - (1.0 punto(s))

Revise el siguiente fragmento y determine qué afirmación es correcta:

```
path("productos/", views.listado)
```

Relación ForeignKey - (1.0 punto(s))

Analice el siguiente código y determine qué representa técnicamente:

```
categoria = models.ForeignKey(Categoria, on_delete=models.CASCADE)
```

Modelos y migraciones - (1.0 punto(s))

Complete correctamente la relación entre modelos y base de datos en Django:

PUT con body - (1.0 punto(s))

Complete correctamente el fragmento para enviar datos en una petición PUT:

```
fetch("/api/usuarios/5", {  
  method: "PUT",  
  headers: {  
    "Content-Type": "application/json"  
  },  
  ____: JSON.stringify({ nombre: "Ana" })  
});
```

GET con parámetros - (1.0 punto(s))

¿Cuál de los siguientes fragmentos construye correctamente una petición HTTP GET con parámetros?

Definición de API - (1.0 punto(s))

Complete correctamente la definición técnica:

"En desarrollo web, una API permite que _____ se comuniquen entre sí sin conocer su implementación interna."

Flujo de comunicación API - (1.0 punto(s))

Seleccione la secuencia correcta del flujo de comunicación cuando se consume una API desde un frontend:

Escenario de uso de API - (1.0 punto(s))

¿Cuál escenario representa un uso correcto de una API en desarrollo web?

Inclusión de templates - (1.0 punto(s))

Seleccione la instrucción correcta para incluir otro template dentro de uno principal:

{% _____ "menu.html" %}

Impresión de variables - (1.0 punto(s))

¿Cuál de los siguientes fragmentos muestra una impresión correcta de una variable enviada desde la vista?

GET y operación CRUD - (1.0 punto(s))

Analice el siguiente endpoint consumido desde JavaScript:

```
fetch("/api/usuarios/8", {  
  method: "GET"  
});
```

¿Qué operación CRUD se ejecuta correctamente?

POST y creación - (1.0 punto(s))

Complete el método HTTP correcto para crear un nuevo recurso:

```
fetch("/api/productos", {  
  method: "_____",  
  headers: { "Content-Type": "application/json" },  
  body: JSON.stringify({ nombre: "Teclado" })  
});
```

PUT y actualización - (1.0 punto(s))

Considere el siguiente fragmento:

```
fetch("/api/clientes/3", {  
  method: "PUT",  
  body: JSON.stringify({ nombre: "Luis", correo: "l@x.com" })  
});
```

¿Cuál afirmación es técnicamente correcta?

PATCH y actualización parcial - (1.0 punto(s))

Complete correctamente el método para una actualización parcial:

```
fetch("/api/pedidos/10", {  
  method: "_____",  
  headers: { "Content-Type": "application/json" },  
  body: JSON.stringify({ estado: "enviado" })  
});
```