

Casos práctico 3 - Examen Primera Unidad

Nombre: Carlos Rivera

NRC: 23307

Caso práctico 3 (Interactivity)

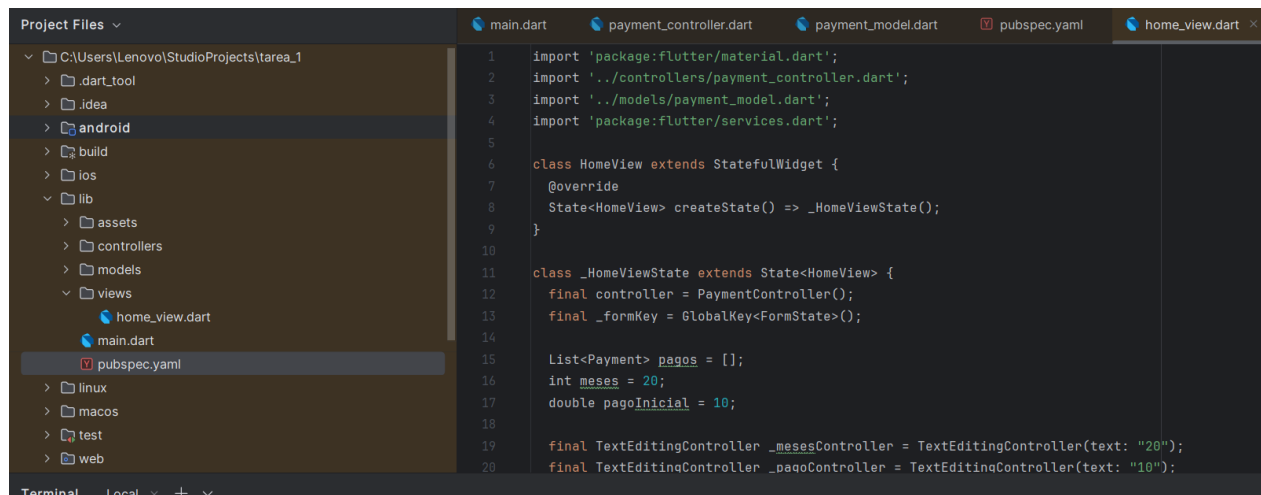
Tema: App "Crédito Fácil"

Descripción: Calcula pagos mensuales con incremento exponencial del importe.

Interactividad:

- Usa GestureDetector e InkWell para interacción visual al seleccionar meses.
- Formulario interactivo para ajustar cantidad de meses o importe.
- Usa InteractiveViewer para gráficos detallados de crecimiento exponencial de pago mensual.

Problema 3: Una persona adquirió un producto para pagar en 20 meses. El primer mes pagó \$10, el segundo \$20, el tercero \$40 y así sucesivamente. Realice un algoritmo para determinar cuánto debe pagar mensualmente y el total de lo que pagó después de los 20 meses y represéntelo mediante el diagrama de flujo, el pseudocódigo y el diagrama N/S utilizando el ciclo apropiado.



```
1 import 'package:flutter/material.dart';
2 import '../controllers/payment_controller.dart';
3 import '../models/payment_model.dart';
4 import 'package:flutter/services.dart';
5
6 class HomeView extends StatefulWidget {
7   @override
8   State<HomeView> createState() => _HomeViewState();
9 }
10
11 class _HomeViewState extends State<HomeView> {
12   final controller = PaymentController();
13   final _formKey = GlobalKey<FormState>();
14
15   List<Payment> _pagos = [];
16   int _meses = 20;
17   double _pagoInicial = 10;
18
19   final TextEditingController _mesesController = TextEditingController(text: "20");
20   final TextEditingController _pagoController = TextEditingController(text: "10");
```

The screenshot shows an IDE with a project file explorer on the left and a code editor on the right. The project file explorer shows a directory structure for an Android project. The code editor displays the `payment_controller.dart` file, which contains the following Dart code:

```
1 import '../models/payment_model.dart';
2
3 class PaymentController {
4   List<Payment> calcularPagos(int meses, double pagoInicial) {
5     List<Payment> pagos = [];
6     double monto = pagoInicial;
7
8     for (int i = 1; i <= meses; i++) {
9       pagos.add(Payment(i, monto));
10      monto *= 2;
11    }
12
13    return pagos;
14  }
15
16  double calcularTotal(List<Payment> pagos) {
17    return pagos.fold(0, (total, p) => total + p.monto);
18  }
19 }
20
```

The screenshot shows a web application titled "Crédito Fácil". It features a form for entering payment data and a list of calculated payments.

Form: Ingrese los datos

Meses	Pago inicial (\$)
12	10.25

Calcular

Results:

- 1 Pago: \$10.25
- 2 Pago: \$20.50
- 3 Pago: \$41.00
- 4 Pago: \$82.00
- 5 Pago: \$164.00
- 6 Pago: \$328.00

Total Pagado: \$41973.75

Link: https://github.com/csrivera3/Caso3_Examen

Calificación:

Criterio	Calificación
Diseño UI: Interfaz clara y sencilla, cumple con mostrar resultados.	1P – 0P
Implementación de Widgets: Usa adecuadamente widgets como Text, Column, Row, Container.	1P – 0P
Correcta lógica del código: El cálculo de ventas por categoría funciona correctamente y es eficiente.	1P – 0P
Funcionalidad: La aplicación ejecuta correctamente mostrando resultados exactos según los datos ingresados.	1P – 0P