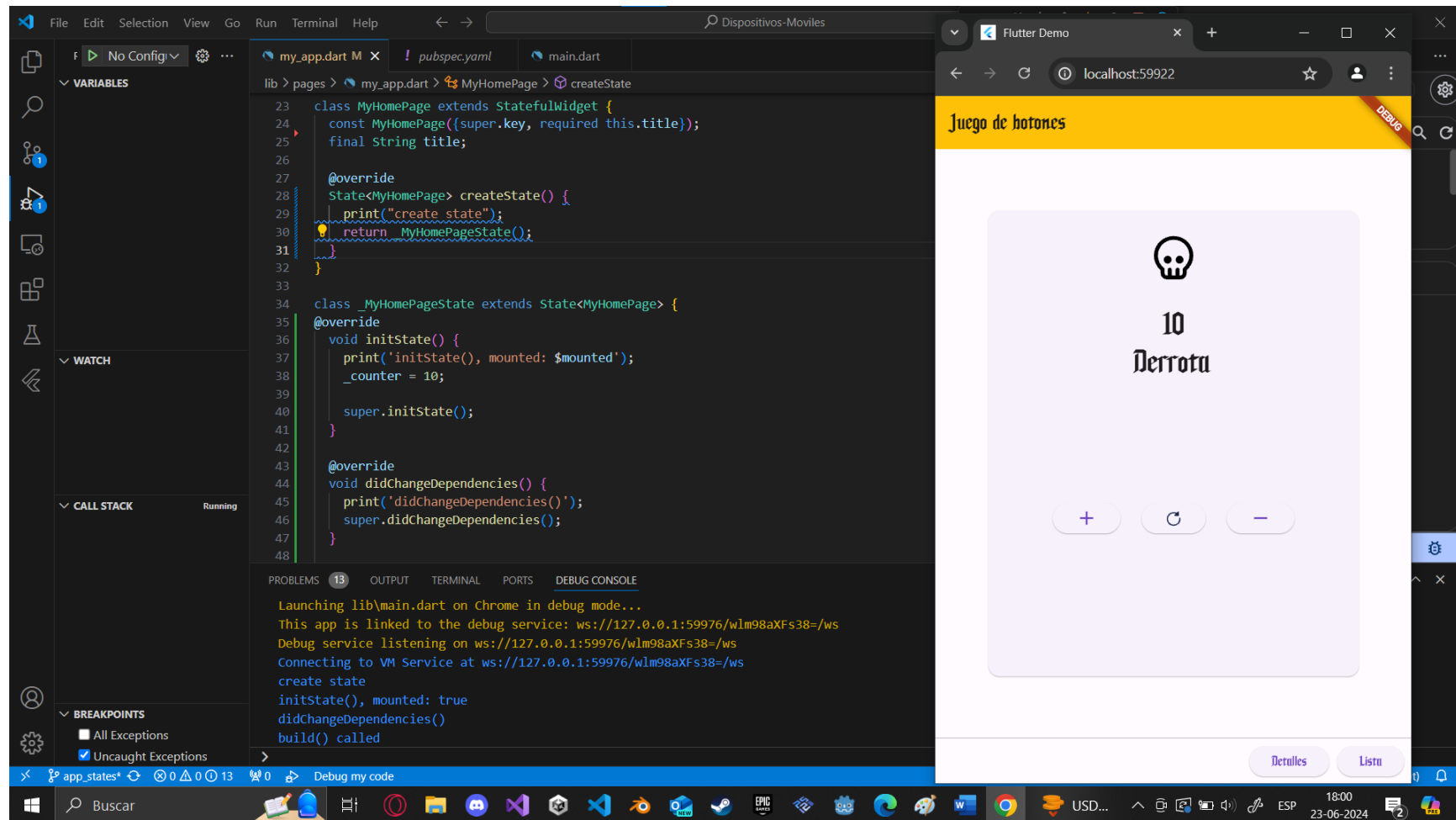


## Mensaje “create state”, initState() y didChangeDependencies()



## didUpdateWidget(), deactivate(), dispose() y reassemble()

The image shows a screenshot of an IDE (Visual Studio Code) with a Flutter project. The code editor displays the `MyHomePageState` class, which implements the `State<MyHomePage>` interface. The code includes the following methods:

```
class _MyHomePageState extends State<MyHomePage> {  
  @override  
  void didUpdateWidget(covariant MyHomePage oldWidget) {  
    print('didUpdateWidget()');  
    super.didUpdateWidget(oldWidget);  
  }  
  
  @override  
  void deactivate() {  
    print('deactivate()');  
    super.deactivate();  
  }  
  
  @override  
  void dispose() {  
    print('dispose()');  
    super.dispose();  
  }  
  
  @override  
  void reassemble() {  
    print('reassemble()');  
    super.reassemble();  
  }  
}
```

The bottom panel shows the **DEBUG CONSOLE** output, which includes the following log messages:

```
Restarted application in 421ms.  
create state  
initState(), mounted: true  
didChangeDependencies()  
build() called  
setState +  
build() called  
reassemble()  
build() called
```

The right panel shows a web browser (Chrome) displaying the Flutter app running on a mobile device. The app has a yellow header with the text "Juego de botones" and a purple button labeled "DEBUG". The main content area displays a skull icon, the number "11", and the word "Derrota" (Defeat). At the bottom, there are three buttons: "+", "↺", and "-".

## setState() y build()

The screenshot displays an IDE interface with the following components:

- Left Sidebar:** Contains icons for Explorer, Search, Run and Debug, Extensions, and a user profile icon.
- Top Panel:** Shows the file explorer with tabs for `my_app.dart`, `pubspec.yaml`, and `main.dart`.
- Editor:** Displays the Dart code for `MyHomePageState`. The code includes methods for incrementing, decrementing, and resetting a counter, each using `setState()` to update the state and `build()` to rebuild the UI. A `print()` statement is used to log when `build()` is called.
- Bottom Panel:** The **DEBUG CONSOLE** tab is active, showing a log of `build()` calls and `setState` updates. The log indicates that `build()` is called multiple times in response to `setState` calls.
- Right Panel:** A preview of the running Flutter app. The app has a yellow header titled "Juego de botones" and a large purple button in the center displaying the number "1". Below the button are two smaller buttons, one with a "+" sign and one with a "-" sign.