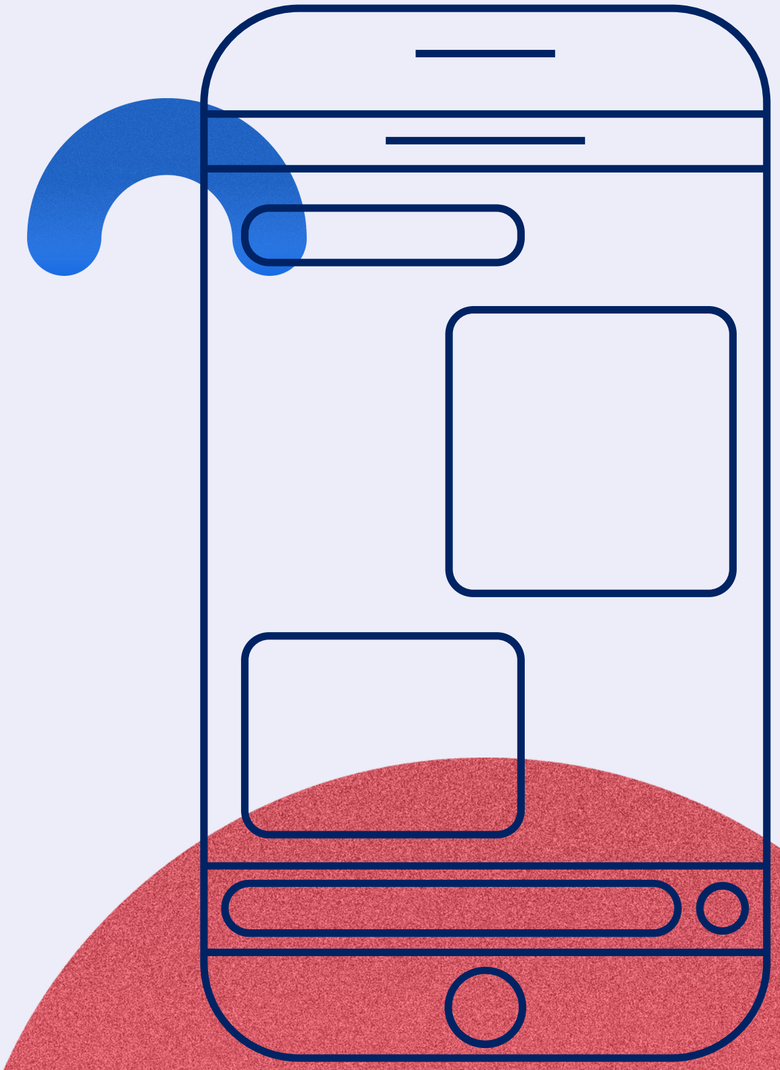
{dev/talles}

¿Qué es Dart?

Tip: Lenguaje de programación

Lenguaje del lado del cliente optimizado para aplicaciones



Optimizado para el UI

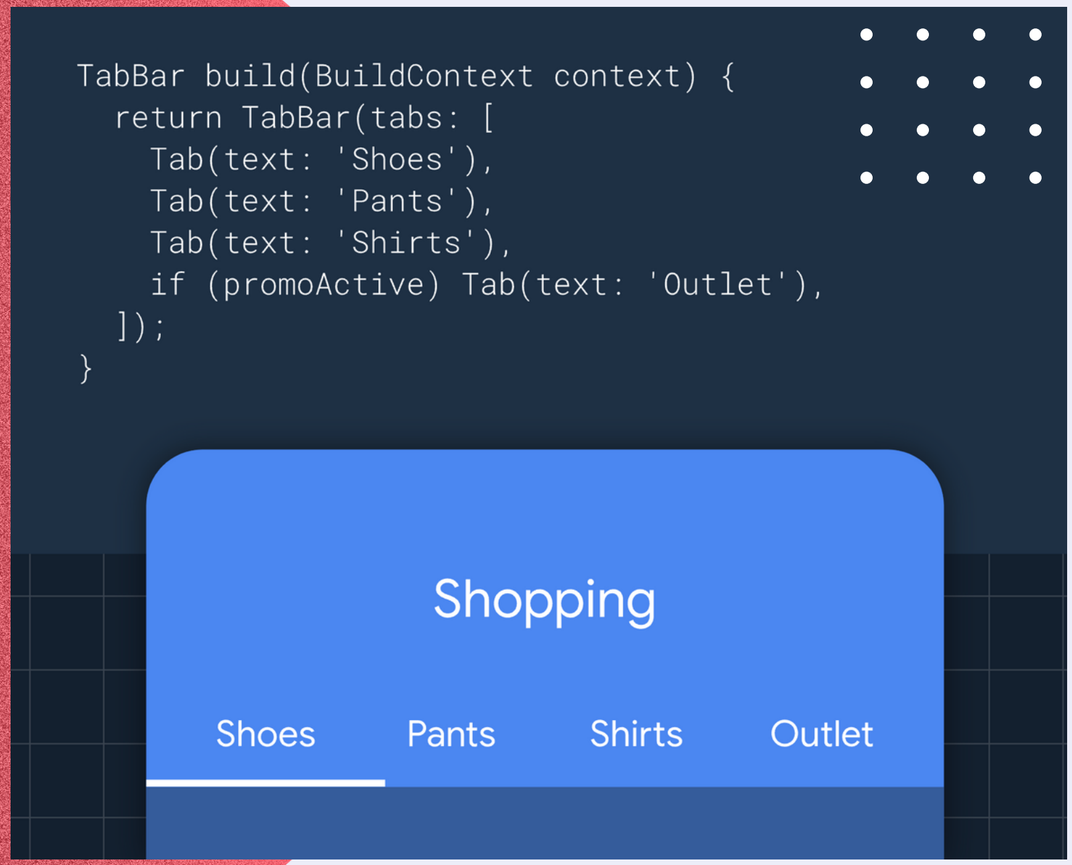
Permite el Hot Reload

Rápido en todas las plataformas (ARM & x64)

devtalles.com

OPTIMIZADO PARA EL UI

```
TabBar build(BuildContext context) {
  return TabBar(tabs: [
    Tab(text: 'Shoes'),
    Tab(text: 'Pants'),
    Tab(text: 'Shirts'),
    if (promoActive) Tab(text: 'Outlet'),
  ]);
}
```

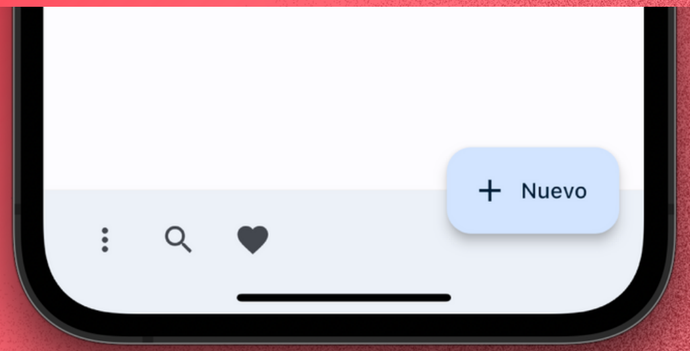


```dart
IconButton(
  icon: const Icon(Icons.more_vert),
  onPressed: () {},
), // IconButton

IconButton(
  tooltip: 'Search',
  icon: const Icon(Icons.search),
  onPressed: () {},
), // IconButton

IconButton(
  tooltip: 'Favorite',
  icon: const Icon(Icons.favorite),
  onPressed: () {},
), // IconButton
```

```dart
floatingActionButton: FloatingActionButton.extended(
  onPressed: () {},
  label: const Text('Nuevo'),
  icon: const Icon(Icons.add),
), // FloatingActionButton.extended
```



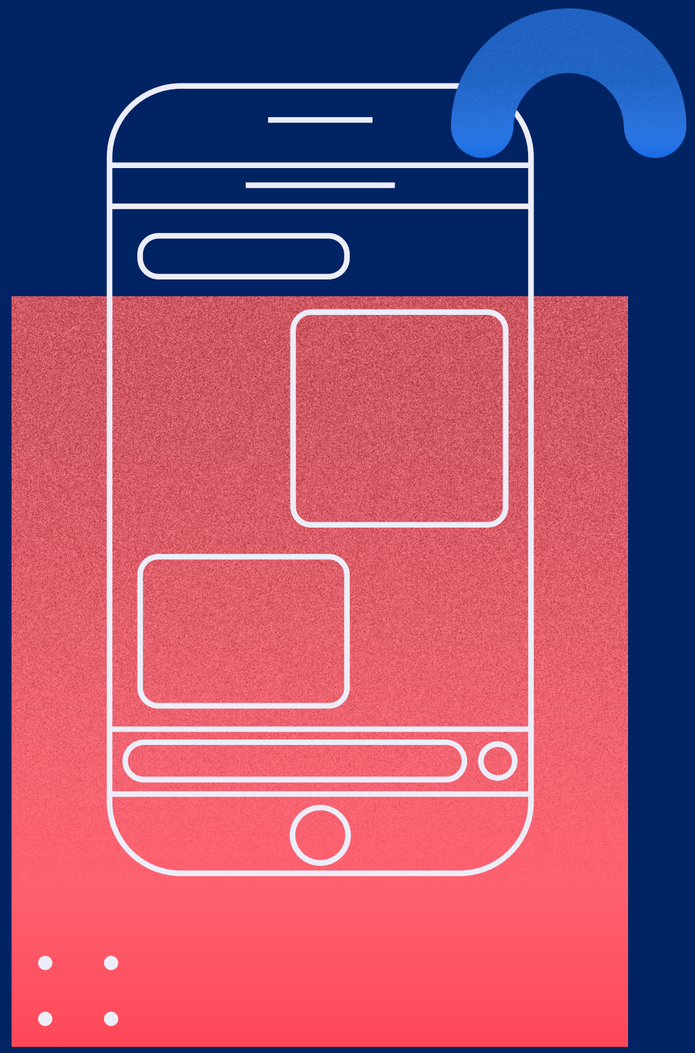
```dart
IconButton(
  icon: const Icon(Icons.more_vert),
  onPressed: () {},
), // IconButton

IconButton(
  tooltip: 'Search',
  icon: const Icon(Icons.search),
  onPressed: () {},
), // IconButton

IconButton(
  tooltip: 'Favorite',
  icon: const Icon(Icons.favorite),
  onPressed: () {},
), // IconButton
```

```dart
floatingActionButton: FloatingActionButton.extended(
  onPressed: () {},
  label: const Text('Nuevo'),
  icon: const Icon(Icons.add),
), // FloatingActionButton.extended
```

CARACTERÍSTICAS



Futures, Async-Await, código non-blocking, Streams al abrirlo de la caja

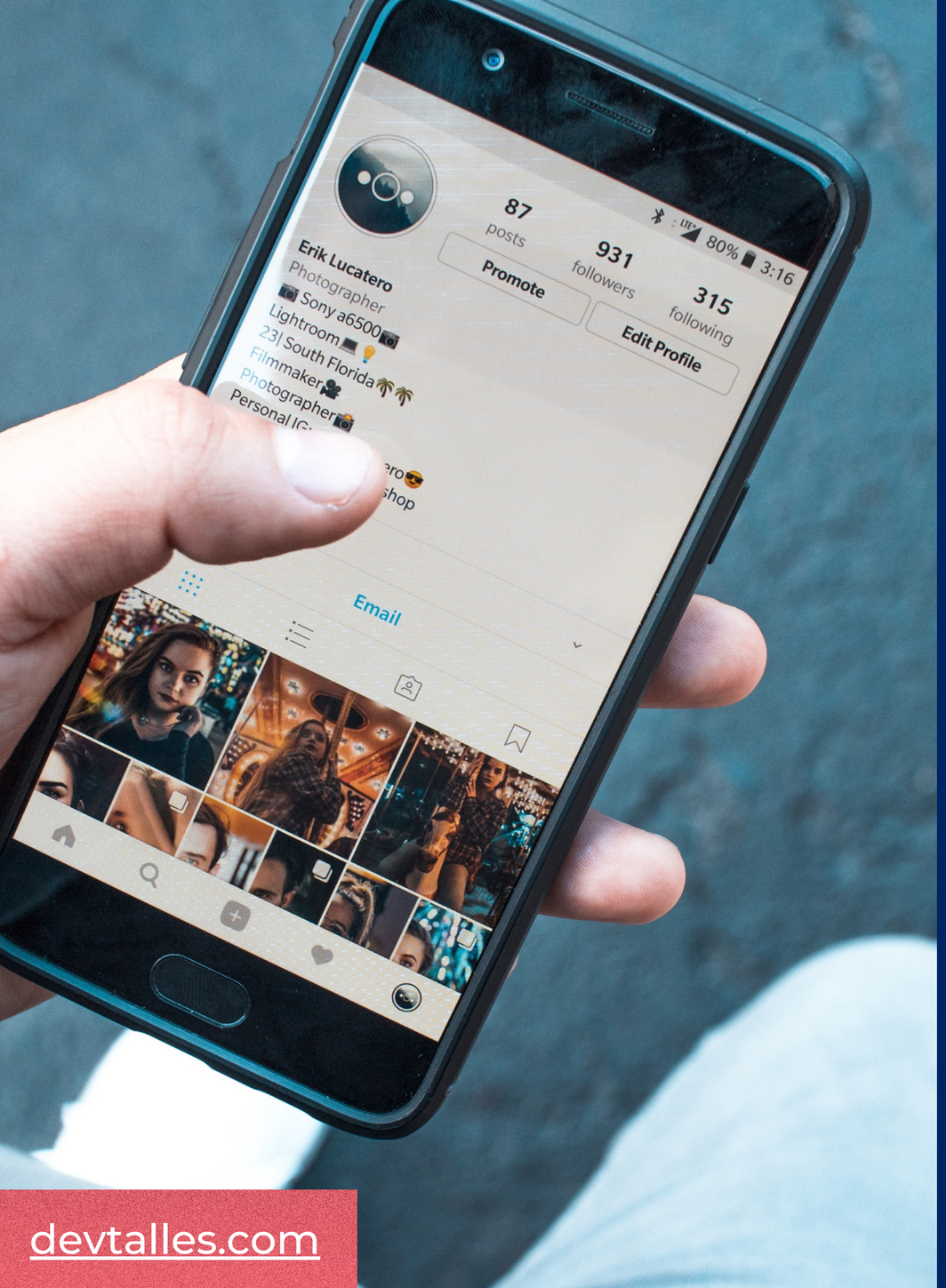
Toda aplicación de Dart ejecuta una función inicial llamada main()

```
Run | Debug | Profile
void main() {

}
```

Sintaxis familiar:
C#, Java, TypeScript

Curva de aprendizaje muy baja*



Flutter

Dart

React Native

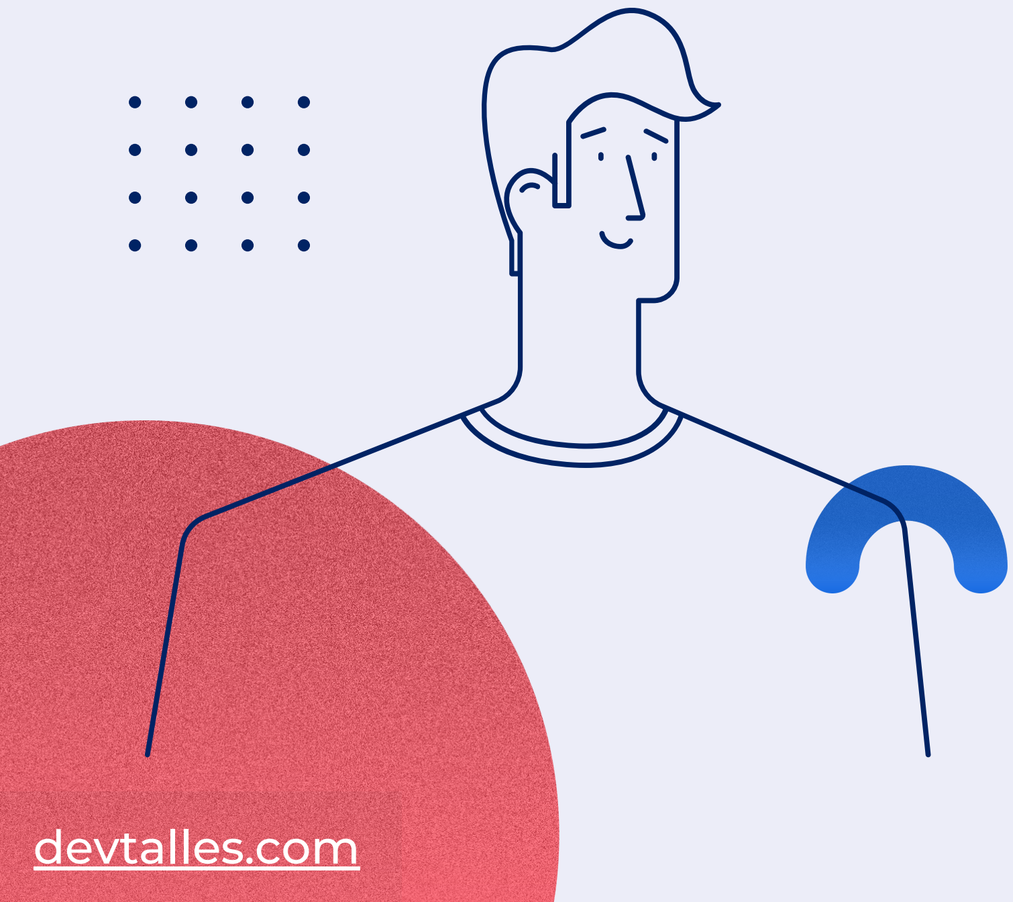
JavaScript

JS-Bridge

Compiler, Aplicación final

¿CÓMO ES POSIBLE ESTO?

VAMOS A COMENZAR



¿Qué es Flutter?

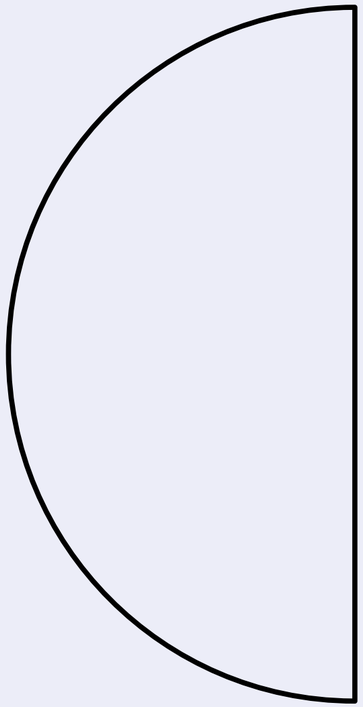
SDK portable - Open source framework - UI Widget Library

Para crear hermosas aplicaciones compiladas de forma nativa, multi-plataforma con un único código base.

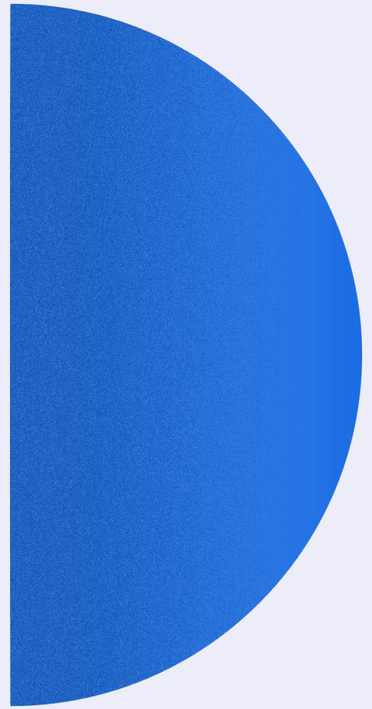
En pocas palabras

Widgets

Username

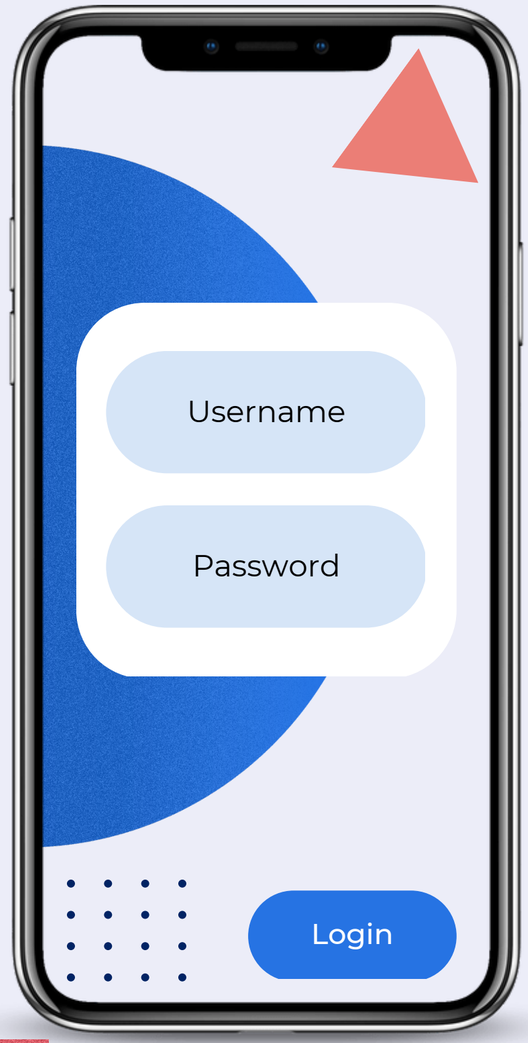


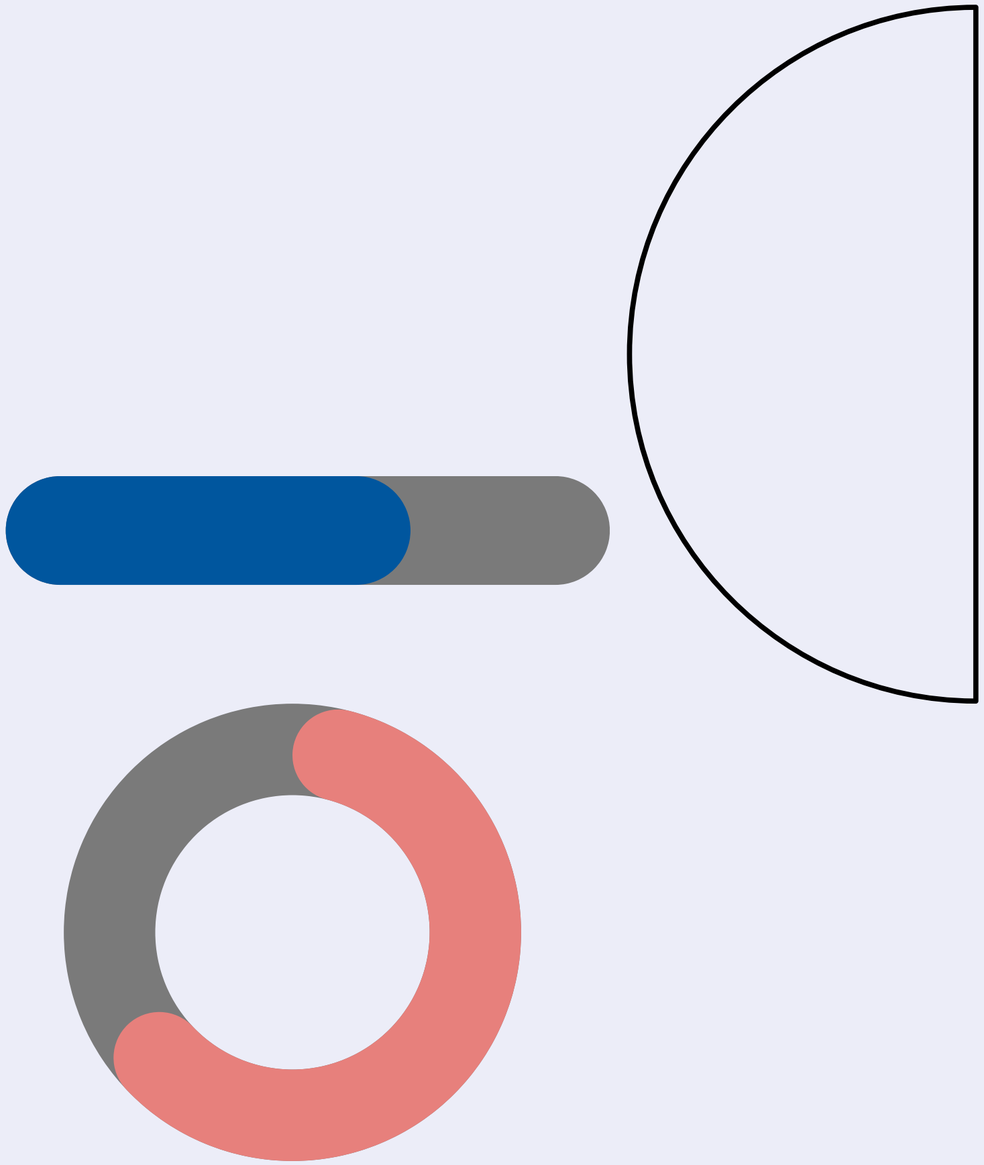
Password



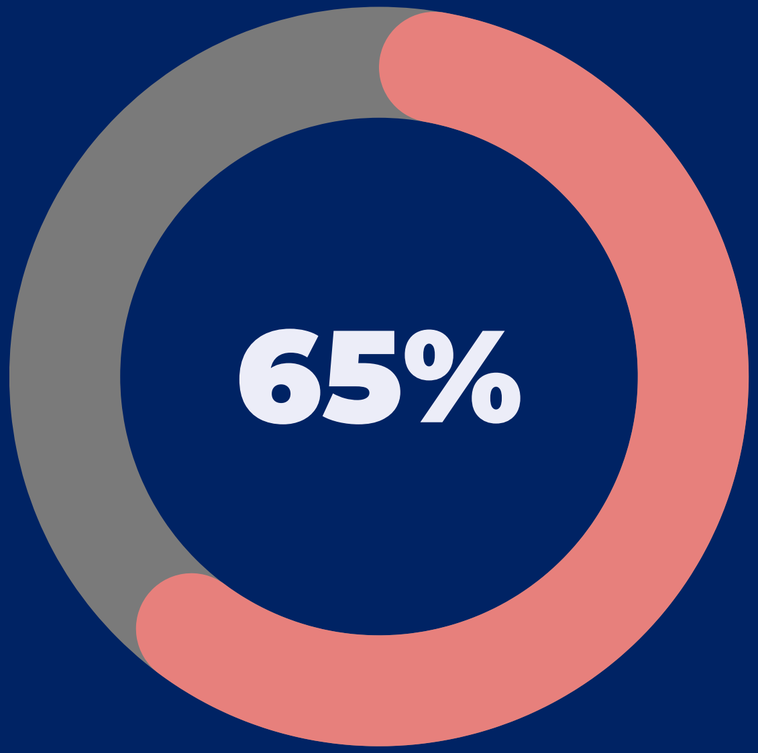
Login

En pocas palabras

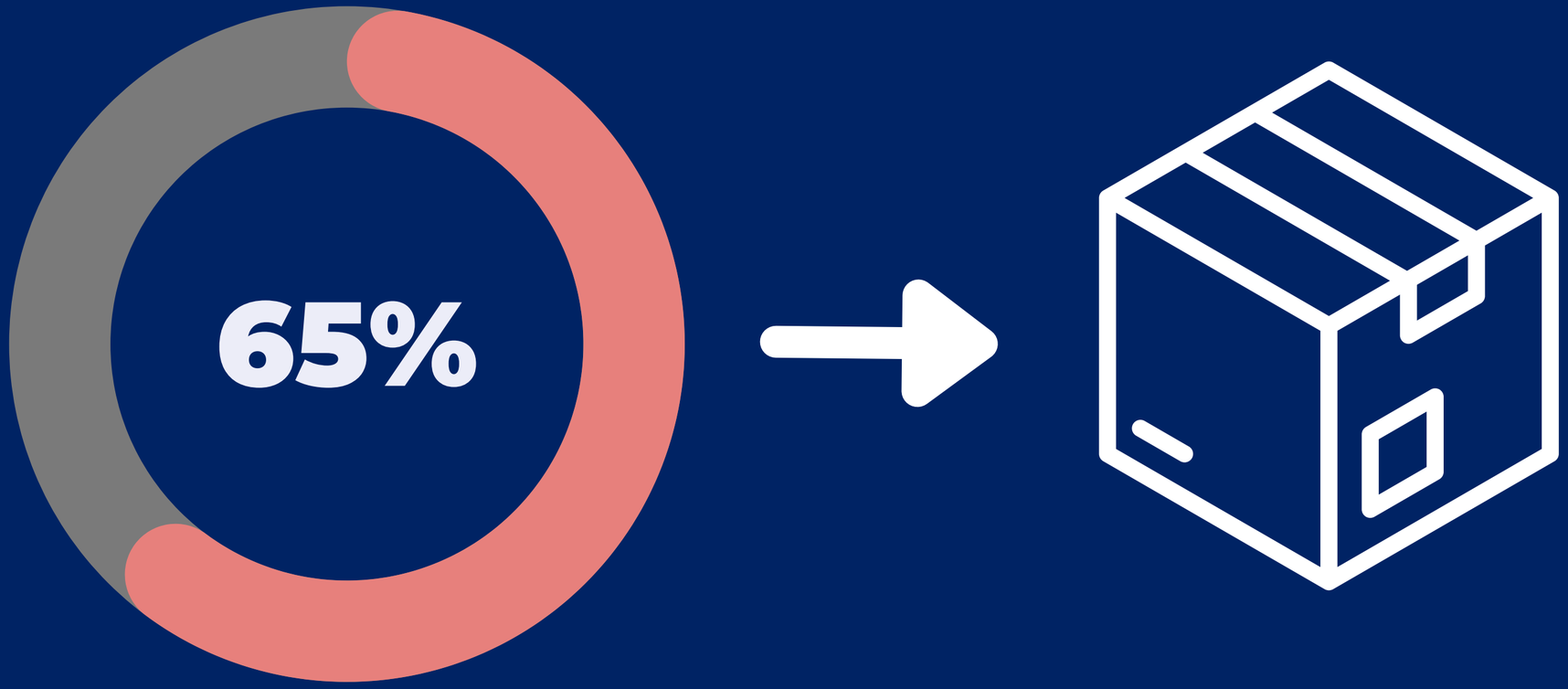




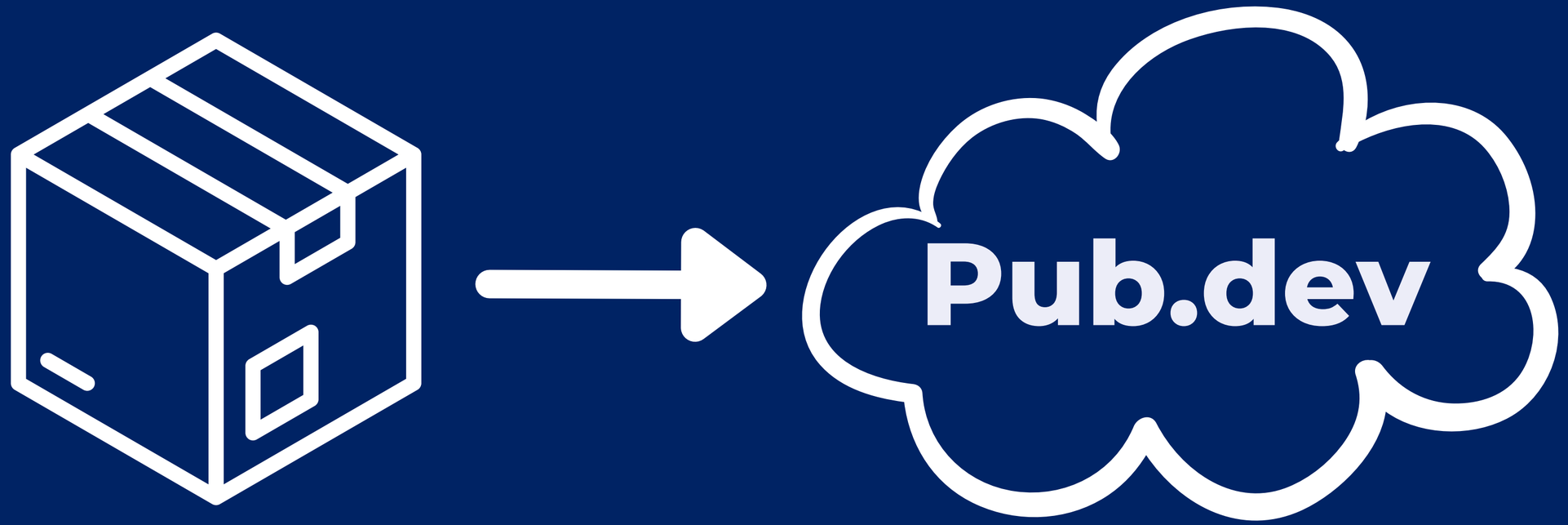
También puedes crear tus propios widgets



También puedes crear tus propios widgets



También puedes crear tus propios widgets



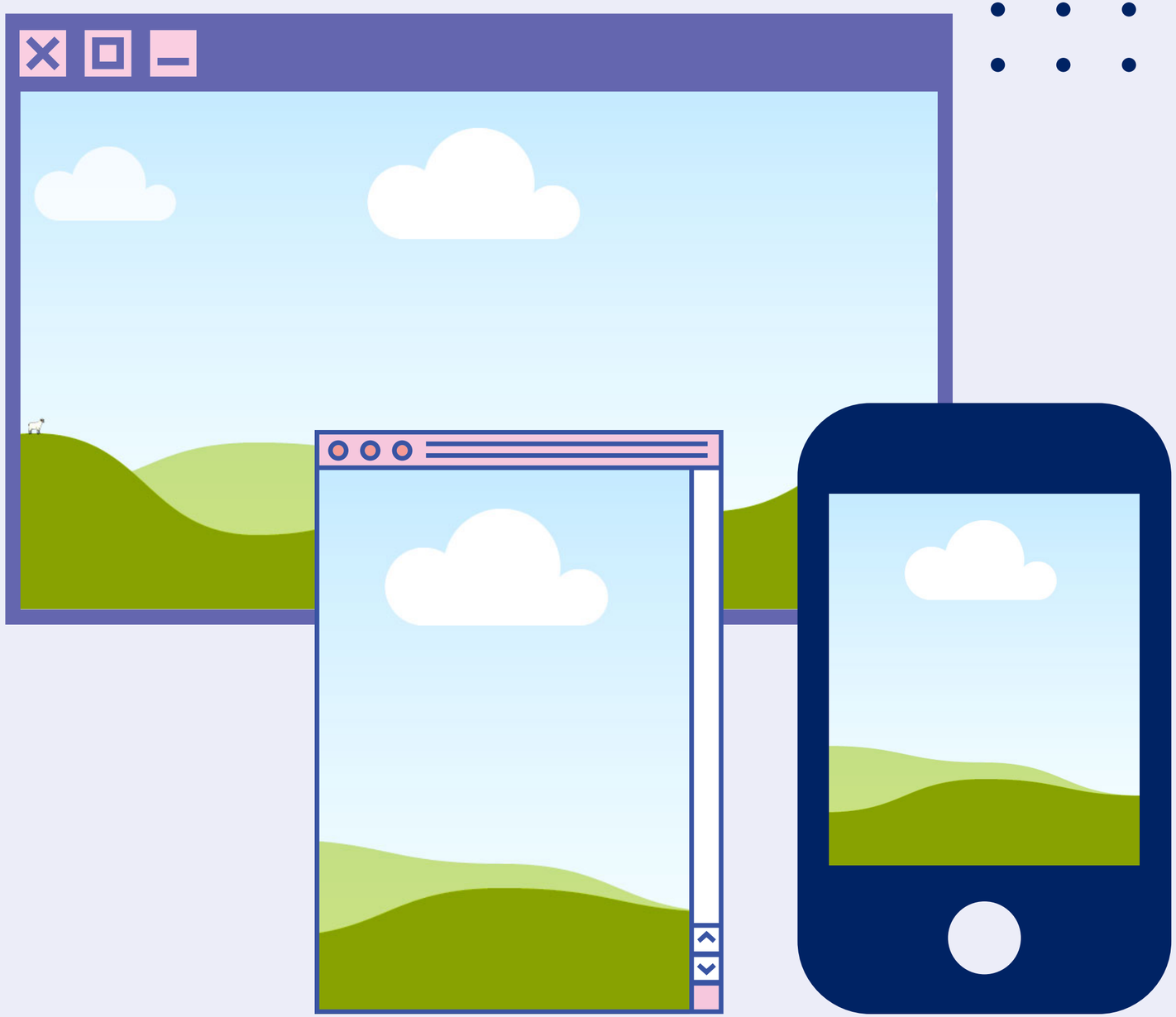
Con una sola base de código

Web

Windows, Linux y Mac

IOS y Android

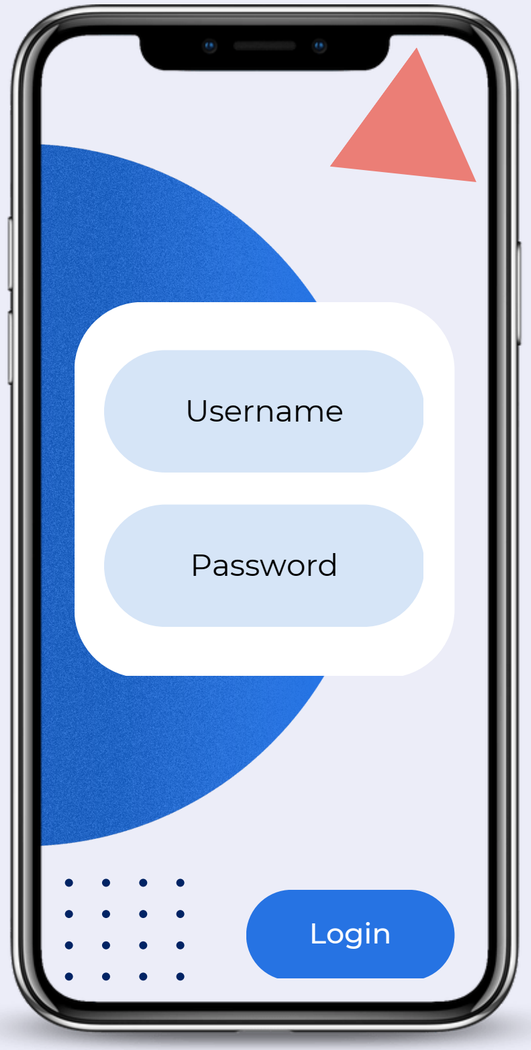
Embebido



¿Deberíamos hacer eso?

Web
Windows, Linux y Mac
IOS y Android
Embebido

Empecemos nuestro camino



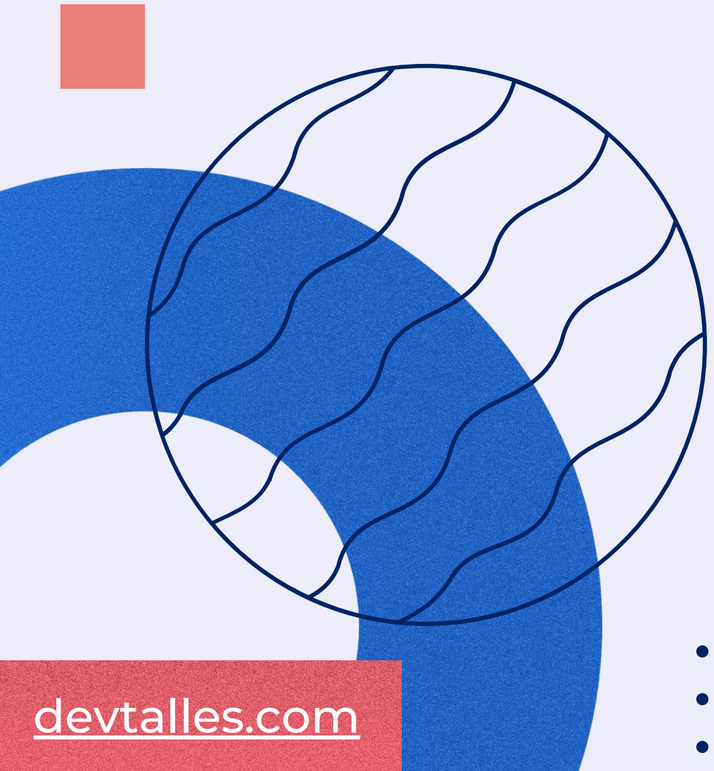
devtalles.com

Continuación veremos

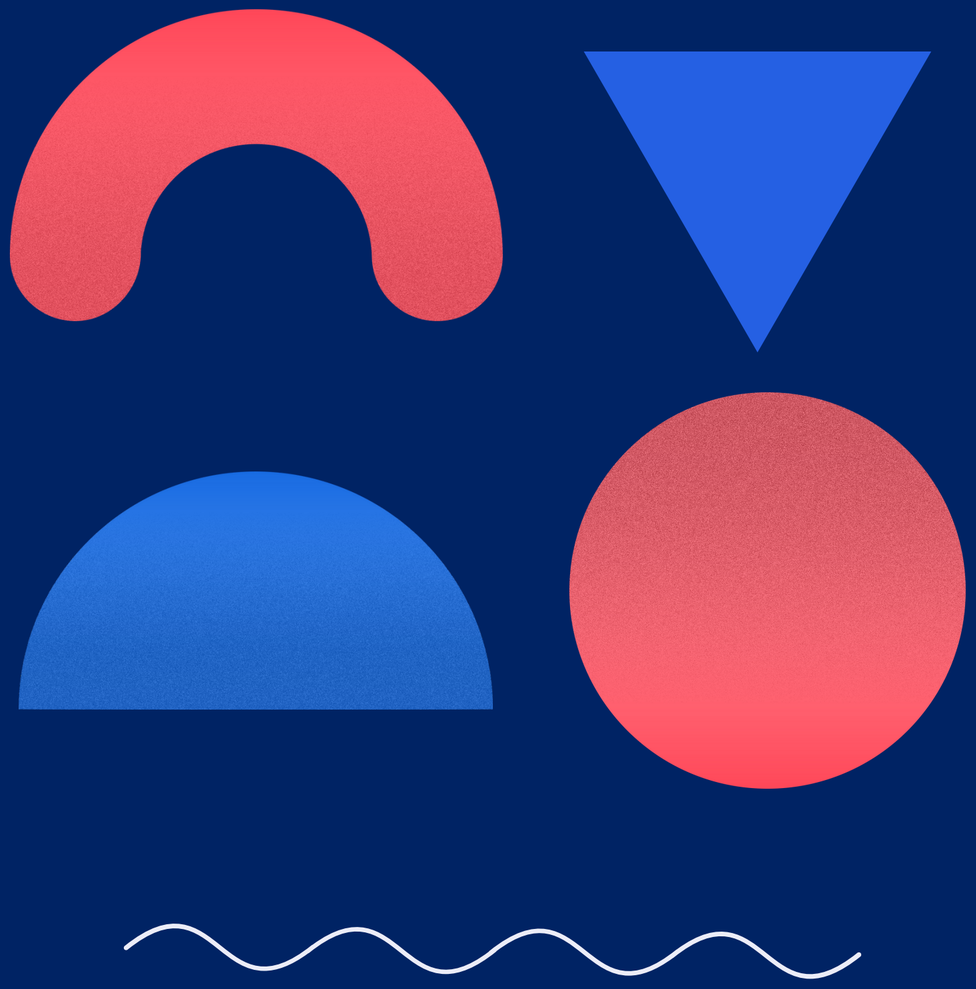
- Entidades
- Datasources
  - Abstractos
  - Implementaciones
- Repositories
  - Abstractos
  - Implementaciones
- Gestor de Estado

ENTIDADES

Clientes
Productos
Películas



Podemos pensar en las entidades como objetos que son y serán idénticos entre diferentes aplicaciones de nuestra empresa.



DATASOURCES

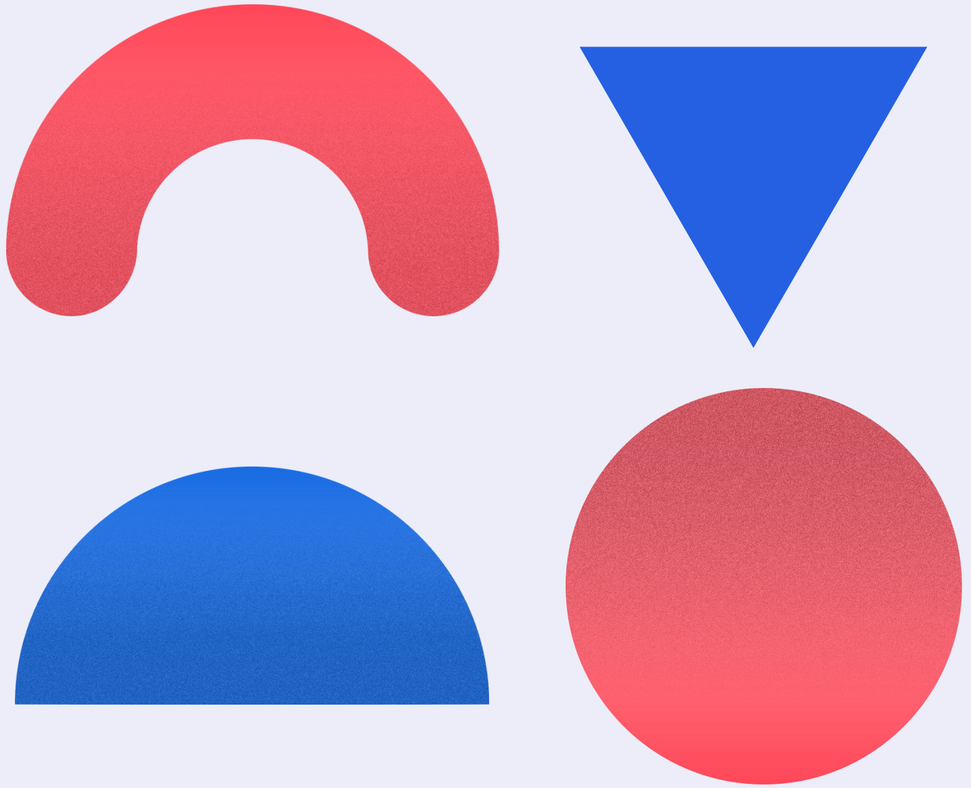
Fuentes de datos.

No debería de importar de dónde venga la data.

Si de TheMovieDB, IMDB, Netflix API, etc.

REPOSITORIOS

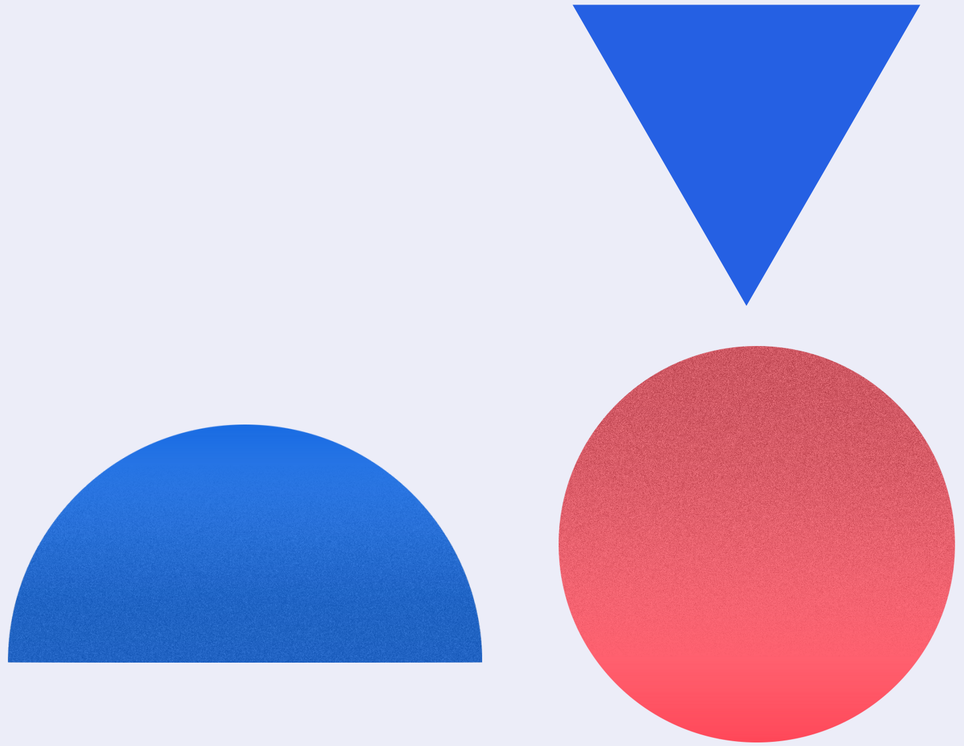
Llaman los origines de datos



Deben de ser flexibles para poder cambiarlos en cualquier momento sin afectar nuestra aplicación

REPOSITORIOS

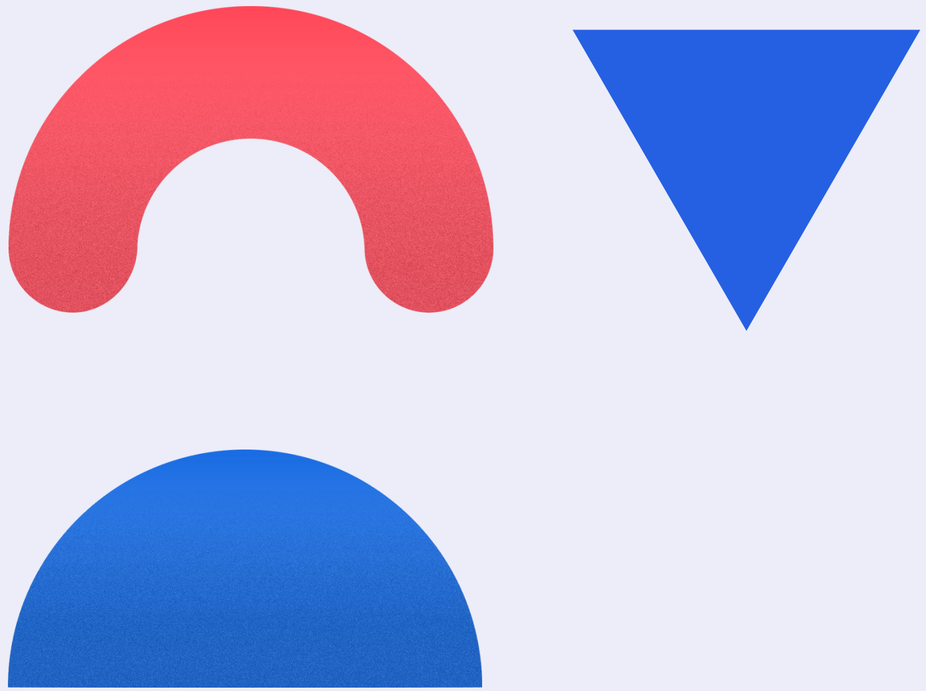
Llaman los origines de datos



Deben de ser flexibles para poder cambiarlos en cualquier momento sin afectar nuestra aplicación

REPOSITORIOS

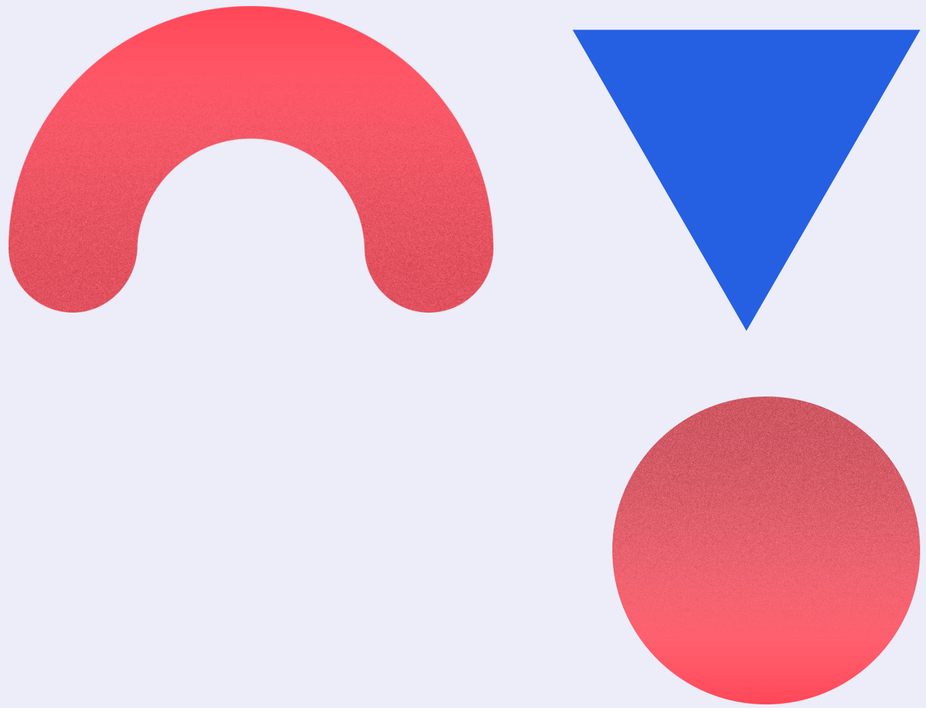
Llaman los origines de datos



Deben de ser flexibles para poder cambiarlos en cualquier momento sin afectar nuestra aplicación

REPOSITORIOS

Llaman los origines de datos



Deben de ser flexibles para poder cambiarlos en cualquier momento sin afectar nuestra aplicación

Gestor de estado

Sirve de puente entre nuestros casos de uso (en este caso el repositorio) y realizan los cambios visuales en los Widgets

En caso de una implementación completa, de arquitectura limpia, el gestor de estado llama casos de uso, y estos al repositorio.

RESUMEN

- Entidades son atómicas
- Los Repositorios llaman Datasources
- Las implementaciones de los Datasources son quienes hacen el trabajo
- El Gestor de estado es el puente que ayuda a realizar los cambios en el UI