

# **TIENDA ONLINE**

**PROYECTO FINAL DE CICLO  
DESARROLLO DE APLICACIONES MULTIPLATAFORMA**

**CURSO 2021-22**



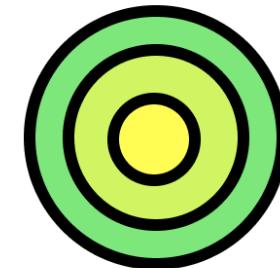
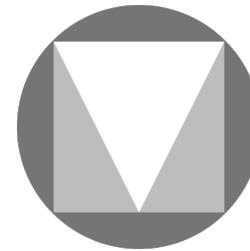
**Alejandro Fandos Ferrada**



**/alejandrfan2**

# TIENDA ONLINE · INSPIRACIÓN





∞ Meta for Developers

# TIENDA ONLINE • Planificación



## MARZO

## ABRIL

## MAYO

## JUNIO

Fase 0	Fase 1	Fase 2	Fase 3	Fase 4	Fase 5	Fase 6	Fase 7	Fase 8
+ Instalación de Java, Flutter,Dart Configuraciones varias de VS Code y GitHub repo. Instalación y configuración de dependencias.	+ Re-factorización directorios. Eliminación de código residual de 'flutter create'.	+ Creación sistema de páginas y navegación. Diseño de las páginas del sistema de autenticación. Diseño del resto de páginas ('Home', 'Cart', 'Profile').	+ Creación del proyecto en Firebase y configuración básica. Creación del Login Service para la autenticación.	+ Creación de entidades para la asimilación de los datos en Flutter. Creación de JSON para Productos / Categorías.	+ Creación de IU de las páginas de cambio de nombre, correo, teléfono, contraseña y avatar. Creación de sistema de cambio de nombre, correo, teléfono, contraseña y avatar.	+ Creación de un sistema de carrito local funcional. Implementación de 'FireBaseStorage' en el carrito.	+ Creación de un sistema de tags para los productos y categorías. Implementación del sistema de productos y tags en 'FireBaseStorage' Creación de un sistema de búsqueda por tags. Creación de IU para la búsqueda.	+ Creación de IU para la selección de coche. Creación de un sistema de selección de coche. Implementación del sistema en 'FireBaseStorage'. Implementar la selección de coche en el sistema de Búsqueda.



# TIENDA ONLINE · SISTEMA DE AUTENTICACIÓN



The screenshot shows the Firebase Authentication interface for the project "CuevaDelRecambio".

**Proveedores de acceso (Access Providers):**

Proveedor (Provider)	Estado (Status)
Correo electrónico/contraseña (Email/Password)	Habilitado (Enabled)
Teléfono (Phone)	Habilitado (Enabled)
Google	Habilitado (Enabled)
Facebook	Habilitado (Enabled)

**Dominios autorizados (Authorized Domains):**

Límite de frecuencia del usuario (User frequency limit): 0 Usuarios limitados (0 Limited users).

**Limitación de frecuencia de la aplicación (App frequency limit):**

Cueva-Auth (App ID: 1005285076793837) - 0 % del límite usado (0% of the limit used). A progress bar shows 100% restante (Remaining).

**Panel (Sidebar):**

- Meta for Developers
- Cueva-Auth (Selected)
- Identificador de la aplicación (App ID)
- Panel (Selected)
- Configuración
- Roles
- Alertas
- Revisión de la aplicación
- Productos (Products)
- Añadir producto (Add product)
- Inicio de sesión con Facebook (Log in with Facebook)
- Registro de actividad (Activity log)



# TIENDA ONLINE • SISTEMA DE AUTENTICACIÓN

```
Future<bool> login(BuildContext context, String email,
String password) async {
    try {
        GetIt.I<FirebaseAuth>()
            .signInWithEmailAndPassword(email: email, password:
password);

        return true;
    } on FirebaseAuthException catch (e) {
        if (e.code == 'weak-password')
            ScaffoldMessenger.of(context).showSnackBar(
                SnackBar(content: Text(
                    "Try using a strong password.")));
        else if (e.code == 'email-already-in-use')
            ScaffoldMessenger.of(context)
                .showSnackBar(SnackBar(content: Text(
                    "That email is in use yet.")));
        else
            ScaffoldMessenger.of(context)
                .showSnackBar(SnackBar(content: Text(
                    "An error occurred.")));
        return false;
    }
}
```

```
final GoogleSignInAccount googleSignInAccount =
    await GoogleSignIn().signIn();

if (googleSignInAccount != null) {
    final GoogleSignInAuthentication googleSignInAuthentication =
        await googleSignInAccount.authentication;

    final AuthCredential credential = GoogleAuthProvider.credential(
        accessToken: googleSignInAuthentication.accessToken,
        idToken: googleSignInAuthentication.idToken,
    );
}
```

```
FacebookAuth instance = FacebookAuth.instance;
final LoginResult result = await instance.login();

if (result.status == LoginStatus.success) {
    final AuthCredential credential =
        FacebookAuthProvider.credential(result.accessToken.token);
```

# TIENDA ONLINE • CARRITO AVANZADO



```
class CartProvider with ChangeNotifier {  
  CartProvider(this._cart) {  
    _cart.forEach((element) {  
      if (element.quantity > 0) {  
        totalPrice += element.quantity * element.price;  
      }  
    });  
  }  
  
  List<Product> _cart;  
  double totalPrice = 0;  
  bool loading = false;  
  
  List<Product> get cart => _cart;  
  ...  
}
```

```
void addProduct(Product product) {  
  int index = _cart.indexWhere((element) => element.id == product.id);  
  
  if (index == -1) {  
    totalPrice += product.price;  
    _cart.add(product);  
  } else {  
    _cart[index].quantity++;  
    totalPrice += product.price;  
  }  
  
  updateCartToFB(_cart);  
  notifyListeners();  
}
```



# TIENDA ONLINE • CARRITO AVANZADO (SYNC)



```
Future<bool> updateCartToFB(List<Product> cart) async {
    User user = FirebaseAuth.instance.currentUser;

    if (user != null) {
        try {
            if (cart.isEmpty) {
                DocumentSnapshot snap = await FirebaseFirestore.instance
                    .collection('users')
                    .doc(user.uid)
                    .get();
                if (snap.exists) {
                    await FirebaseFirestore.instance
                        .collection('users')
                        .doc(user.uid)
                        .update({
                            'ids': FieldValue.delete(),
                            'quantities': FieldValue.delete()
                        });
                }
                return true;
            }
        ...
    }
```

```
...
List<int> ids = <int>[];
List<int> quantities = <int>[];

cart.forEach((element) {
    print(
        element.id + ' ' + element.quantity.toString() + ' added to cart');
    ids.add(parseInt(element.id));
    quantities.add(element.quantity);
});

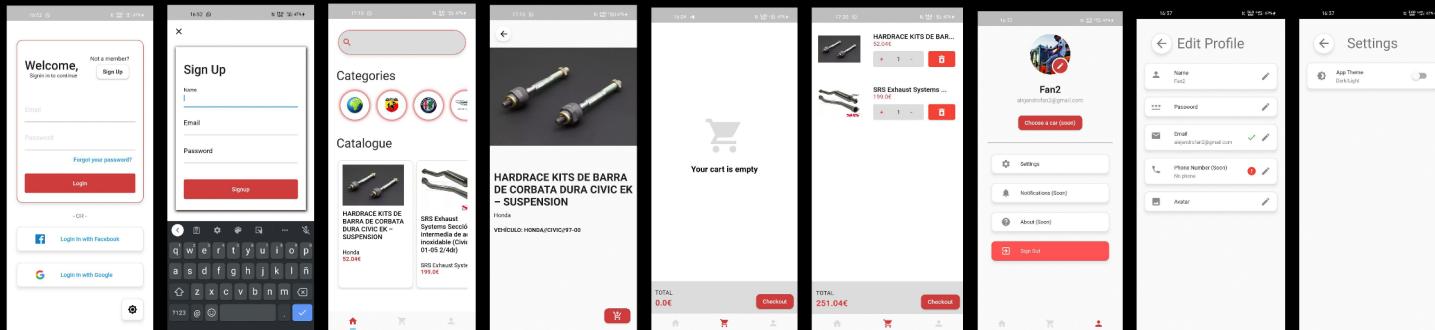
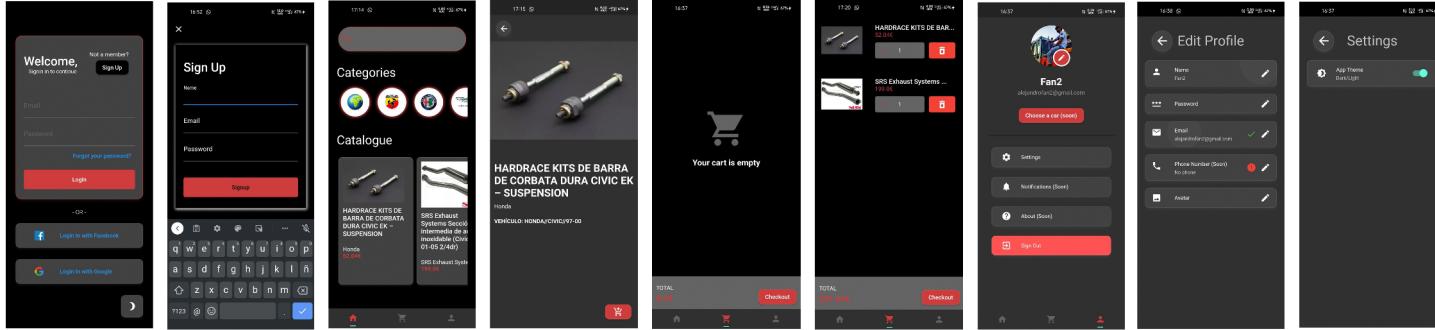
await FirebaseFirestore.instance
    .collection('users')
    .doc(user.uid)
    .set({'ids': ids, 'quantities': quantities});
} on Exception catch (e) {
    print('$e');
    return false;
}
return true;
}
return false;
}
```



# TIENDA ONLINE • DEMO FUNCIONALIDAD



android



# TIENDA ONLINE · PREGUNTAS



# **TIENDA ONLINE**

**PROYECTO FINAL DE CICLO  
DESARROLLO DE APLICACIONES MULTIPLATAFORMA**

**CURSO 2021-22**



**Alejandro Fandos Ferrada**



**/alejandrfan2**