



SELL IT EASY

Curso oficial de desarrollo de aplicaciones móviles
para IOS

David Márquez Delgado
d.marquez.delgado.91@gmail.com

Tabla de contenido

INTRODUCCIÓN	3
DESCRIPCIÓN	4
ARQUITECTURA	5
INTERFAZ DE USUARIO	5
COMPONENTES GENÉRICOS	5
PROTOCOLOS	6
SERVIDOR	6
BASE DE DATOS	6
FASES DE EJECUCIÓN	7
TOMA DE REQUISITOS	7
DISEÑO	7
DESARROLLO	7

Introducción

El objetivo del presente documento es presentar brevemente el proyecto desarrollado para el curso de CICE de desarrollo de aplicaciones móviles para IOS.

Descripción

El proyecto consiste en el desarrollo de una aplicación móvil para IOS 10.0 que permita gestionar un conjunto de establecimientos dedicados a la venta en el sector textil.

Arquitectura

El proyecto está dividido en varias capas:

- Interfaz de usuario, desarrollada en Swift para IOS 10.0
- Servidor, desarrollada en Java 1.8 (Hibernate,Maven,Spring)
- Base de datos relacional MySQL.

Interfaz de usuario

Para el desarrollo de la interfaz de usuario ha sido necesario el uso de librerías de terceros. Estas librerías se recogen en el fichero "Podfile" incluido en el proyecto y han sido descargadas desde CocoaPods.

- SwiftyJSON. Traducir contenido de comunicaciones REST.
- ReachabilitySwift. Importada pero sin uso real.
- ExpandingMenu. Menu expandible
- CryptoSwift. Encriptamiento de datos. (importada pero sin uso real)
- PromiseKit. Peticiones REST.
- PKHUD. Animaciones de espera.
- Kingfisher. Descarga de imágenes. (importada pero sin uso real)
- Parse. Acceso BBDD (importada pero sin uso real)
- CarbonKit. Menús animados
- Charts. Gráficos de estadísticas
- FSCalendar. Calendario animado.
- ObjectMapper. Traducción de JSON a clases Swift.

Además se han utilizado iconos descargados de:

<https://thenounproject.com/>

Y música y vídeo descargado desde Youtube:

<https://www.youtube.com/watch?v=-0kcet4aPpQ>

El proyecto está construido bajo el paradigma Modelo-Vista-Controlador (MVC) y se divide en las siguientes carpetas

- ENTITIES. Entidades de intercambio entre el servidor y la interfaz gráfica
- SERVICES. Clientes REST de acceso al servidor.
- VIEW_CONTROLLERS. Controladores
- UTILS. Funciones globales de uso reiterado en el aplicativo.

Componentes genéricos

Se han implementado diferentes componentes genéricos para agilizar el desarrollo del proyecto. Algunos de ellos son:

- Tabla y celdas genéricas.
- Selector de fechas.
- Selector de imágenes.
- Selector de direcciones (Mapa).
- Selector de opciones.
- Editor de texto.

Protocolos

Para cada uno de los componentes genéricos se ha definido un protocolo en Swift que permita trabajar con él.

Servidor

El servidor también ha sido desarrollado bajo el paradigma MVC y su estructura es la siguiente:

- Entidades: Representan las tablas de la base de datos.
- Repositorios: Permiten consultas a la base de datos.
- Controladores: API REST implementada para este proyecto.

Base de datos

La base de datos ha sido creada haciendo uso del ORM Hibernate. Su estructura se adjunta junto al proyecto "EER Diagrama.png".

Cabe destacar que las imágenes en todo momento son tratadas como texto y ha sido necesario configurar MySQL para aceptar transacciones superiores a los 20MB. Dicha configuración se incluye en el proyecto en el fichero "my.cnf"

Fases de ejecución

Toma de requisitos

Previamente al desarrollo se han decidido acotar el proyecto bajo unos requisitos iniciales recogidos en el documento "01. Primeras ideas.docx"

Diseño

Tras la toma de requisitos se ha realizado un diseño a alto nivel de la estructura que compondría el proyecto.

Desarrollo

Para el desarrollo se ha hecho uso de la metodología ágil SCRUM con 3 sprints.

En el primer sprint se ha desarrollado:

- La base de datos.
- Servidor: Entidades y repositorios
- Interfaz: Componentes genéricos y animaciones.

En el segundo sprint:

- Servidor: Principales controladores.
- Interfaz: Implementación de principales funciones. A saber:
 - Registro.
 - Inicio de sesión.
 - Alta de tienda.
 - Consulta de tiendas.

En el tercer y último spring:

- Servidor: Resto de controladores.
- Resto de ventanas: Se han reutilizado los componentes genéricos para todas ellas.