

# OnePercent App - Informe del Estado del Código

## Análisis Técnico Simplificado

**Fecha:** 18 de Octubre, 2025

**Proyecto:** OnePercent - Aplicación de Salud y Bienestar

**Plataforma:** Aplicación móvil (iOS + Android) + Servidor API

### Resumen Ejecutivo

Este documento presenta el estado actual del código de OnePercent de forma clara y directa. Describe **qué tiene el proyecto hoy, qué funciona bien y qué necesita atención**.

### Visión General

#### ¿Qué es el proyecto?

OnePercent es una aplicación móvil de salud y bienestar con:

- **Aplicación móvil** para iOS y Android
- **Servidor backend** que gestiona datos y lógica de negocio
- **Sistema de autenticación** robusto
- **Módulos de salud:** nutrición, ejercicio, seguimiento de actividad

#### Tamaño del Proyecto

Componente	Descripción	Tamaño
Servidor (Backend)	Código que maneja datos y lógica	~15,000 líneas de código
App Móvil (Frontend)	Interfaz de usuario	~20,000 líneas de código
Total	Proyecto completo	~35,000 líneas de código

**Contexto:** Un proyecto de este tamaño es mediano-grande. Es comparable a una app madura con funcionalidades completas.

### Lo Que Funciona Bien

#### 1. Sistema de Login Completo

La aplicación permite a los usuarios registrarse e iniciar sesión de múltiples formas:

##### Email y Contraseña

- Los usuarios pueden crear cuenta con su email
- Las contraseñas están encriptadas (seguras)
- Sistema de validación de email funcional

##### Login con Google

- Los usuarios pueden entrar con su cuenta de Google
- Integración completa y funcional

##### Login con Apple

- Los usuarios pueden entrar con su Apple ID
- Obligatorio para apps iOS, ya implementado

## ✔ Login con Facebook

- Los usuarios pueden entrar con Facebook
- Integración completa

**Estado:** ✔ Completamente funcional y seguro

## 2. Organización del Código

El código está bien estructurado en módulos separados:

### Servidor:

- 16 módulos independientes (usuarios, recetas, actividades, etc.)
- Cada módulo tiene su propia responsabilidad
- Código ordenado y fácil de encontrar

### App Móvil:

- Páginas públicas (login, registro)
- Páginas privadas (perfil, actividades, estadísticas)
- 22 componentes reutilizables
- Bien organizado

**Beneficio:** Un equipo nuevo puede entender el código fácilmente.

## 3. Funcionalidades de la App

La aplicación tiene todas las funcionalidades principales implementadas:

### ✔ Gestión de Usuarios

- Perfiles completos
- Datos de salud (peso, altura, edad)
- Objetivos personalizados

### ✔ Recetas y Nutrición

- Base de datos de recetas
- Información nutricional
- Filtros por alergias e intolerancias
- Recetas favoritas

### ✔ Actividades Físicas

- Registro de ejercicios
- Seguimiento de pasos
- Cápsulas de entrenamiento
- Estadísticas de progreso

### ✔ Sistema de Licencias

- Planes básicos y premium
- Integración con pagos (RevenueCat)
- Renovación automática

### ✔ Multiidioma

- Español e Inglés
- Todas las pantallas traducidas




## 4. Tecnología Moderna

El proyecto usa herramientas y frameworks actualizados:

- Las librerías están al día
- No hay tecnología obsoleta
- Compatible con las últimas versiones de iOS/Android

## 5. Seguridad de Contraseñas

Las contraseñas de los usuarios están:




-  Encriptadas (nadie puede verlas)
  -  Con validación fuerte (mínimo 8 caracteres, mayúsculas, minúsculas)
  -  Nunca se guardan en texto plano
- 

## Áreas Que Necesitan Atención

### 1. Ausencia de Tests Automatizados

¿Qué son los tests? Programas que verifican automáticamente que el código funciona correctamente.

Estado actual:

-  0 tests en el servidor
-  0 tests en la app móvil
-  0% de cobertura

¿Por qué es importante?

- Sin tests, cada cambio puede romper algo sin que nos demos cuenta
- Los bugs pueden llegar a producción fácilmente
- Es difícil saber si una modificación afecta otras partes

**Analogía:** Es como conducir sin cinturón de seguridad. Funciona hasta que hay un accidente.

### 2. Configuración de Seguridad Abierta

**Problema encontrado:** El servidor acepta peticiones desde **cualquier sitio web**, sin restricciones.

¿Por qué es un problema?

- Cualquier página web maliciosa podría intentar acceder
- No hay control de quién puede hacer peticiones
- Vulnerable a ataques externos

**Estado:** Actualmente abierto a todo el internet.

**Analogía:** Es como dejar la puerta de tu casa sin cerradura.

### 3. Modificación Automática de Base de Datos

**Problema encontrado:** El sistema está configurado para modificar automáticamente la estructura de la base de datos.

¿Por qué es un problema?

- Puede borrar datos accidentalmente
- Cambios automáticos en producción son peligrosos
- No hay control manual de cambios críticos

**Estado:** Activado sin verificar el entorno.

**Analogía:** Es como dejar que un robot reorganice tu casa sin supervisión.

### 4. Límites de Subida Muy Altos

**Problema encontrado:** La app permite subir archivos de hasta 5GB (5000 megabytes).

### ¿Por qué es un problema?

- Un usuario podría saturar el servidor
- Costos de almacenamiento innecesarios
- Vulnerable a ataques de denegación de servicio

**Estado:** Configurado en 5000MB (excesivo).

**Analogía:** Es como permitir que alguien llene tu garaje con 100 autos cuando solo necesitas espacio para 1.

## 5. Logs de Depuración en Código ⚠️

**Problema encontrado:** Hay 35+ mensajes de depuración ( `console.log` ) en el código del servidor.

### ¿Por qué es un problema?

- Información sensible puede filtrarse en logs
- Dificulta encontrar errores reales
- No es profesional en producción

**Estado:** 35 instancias en 11 archivos.

## 6. Búsquedas Sin Límite ⚠️

**Problema encontrado:** Cuando un usuario busca recetas o usuarios, el sistema puede devolver miles de resultados sin límite.

### ¿Por qué es un problema?

- La app se puede poner lenta
- Consume mucha memoria
- Mala experiencia de usuario

**Estado:** No hay paginación ni límites.

**Analogía:** Es como pedir una lista de todas las personas en Facebook en vez de ver 10 por página.

## 7. Claves Públicas en el Código ⚠️

**Problema encontrado:** Las claves de Firebase (servicio de Google) están escritas directamente en el código.

### ¿Por qué es un problema?

- Cualquiera que vea el código puede ver las claves
- Aunque es común en apps móviles, es mejor usar configuración

**Estado:** Claves visibles en archivos de configuración.

---

## Estado de la Base de Datos

### Estructura de Datos

La base de datos tiene una estructura completa y bien diseñada:

#### Tablas Principales:



Tabla	Descripción	Relaciones
User	Datos de usuarios	16 relaciones con otras tablas
Recipe	Recetas de comida	Ingredientes, pasos, información nutricional
Ingredient	Ingredientes	Información nutricional completa

<b>UserActivity</b>	Actividades físicas	Usuario, tipo, intensidad, duración
<b>UserLicense</b>	Licencias activas	Usuario, tipo de plan, fechas

## Índices de Base de Datos

¿Qué son los índices? Son como el índice de un libro: ayudan a encontrar información más rápido.

**Estado actual:**

-  Email de usuario tiene índice (búsqueda rápida)
-  Otros campos importantes no tienen índice (búsqueda lenta)

**Impacto:** Algunas búsquedas pueden ser lentas con muchos usuarios.

---



## Estado de Seguridad

### Lo Que Está Bien Protegido

#### Contraseñas Encriptadas

- Imposible ver contraseñas de usuarios
- Algoritmo de encriptación fuerte (bcrypt)

#### Tokens de Sesión

- Sistema de tokens temporales para mantener sesión
- Tokens que expiran automáticamente
- Sistema de renovación automática




#### Validación de Email

- Los usuarios deben validar su email antes de usar la app
- Previene cuentas falsas

#### Protección de Rutas

- Solo usuarios autenticados pueden acceder a contenido privado
- Verificación en cada petición

### Lo Que Necesita Mejora

 **Configuración de CORS** (ver sección anterior)  **Sincronización de BD** (ver sección anterior)  **Límites de Request** (ver sección anterior)

---



## Características de la App Móvil

### Funcionalidades Nativas Implementadas

La app aprovecha las capacidades del teléfono:

#### Cámara

- Los usuarios pueden tomar fotos de perfil
- Fotos de recetas

#### Datos de Salud

- Lee pasos del teléfono automáticamente
- Integración con Health (iOS) / Google Fit (Android)

#### Notificaciones Push

- La app puede enviar notificaciones
- Recordatorios de actividad, etc.

#### ✅ Pagos In-App

- Compra de planes premium dentro de la app
- Integrado con Apple Pay / Google Pay

#### ✅ Vibración (Haptics)

- Feedback táctil en acciones importantes

### Compatibilidad

✅ **iOS:** Funciona en iOS 14+   ✅ **Android:** Funciona en Android 5.0+   ✅ **Soporte:** ~95% de dispositivos actuales

---

## Calidad del Código

### Legibilidad

¿Es fácil de leer el código?

#### ✅ Nombres descriptivos

- Variables y funciones tienen nombres que se entienden
- Ejemplo: `getUserById` en vez de `gubi`

#### ✅ Código organizado

- Archivos separados por función
- Cada cosa en su lugar

#### ⚠️ Comentarios

- Algunos comentarios explicativos
- Podría tener más documentación
- ~10 bloques de código comentado sin eliminar

#### ⚠️ Funciones largas

- Algunas funciones tienen más de 70 líneas
- Deberían dividirse en funciones más pequeñas

### Consistencia

#### ✅ Naming conventions

- Los archivos siguen un patrón consistente
- Fácil encontrar qué hace cada archivo

#### ⚠️ Nombres de directorios

- El directorio del backend se llama "frontend" (invertido)
  - El directorio del frontend se llama "backend" (invertido)
  - Puede confundir a desarrolladores nuevos
- 

## Rendimiento

### Velocidad de la App

Estado actual:

✅ **App móvil responde rápido** en condiciones normales   ⚠️ **Puede ser lento con muchos datos** (sin optimizaciones)

Optimizaciones Presentes

✅ **Compresión de imágenes** (con Sharp) ✅ **Cache en el móvil** (Ionic Storage)

Optimizaciones Ausentes

❌ **Lazy loading** - No hay carga diferida de módulos ❌ **Virtual scrolling** - Listas largas pueden ser lentas ❌ **Paginación** - Las búsquedas no limitan resultados ❌ **CDN para imágenes** - Las imágenes se sirven desde el servidor

**Impacto:** Con pocos usuarios no se nota, pero puede ser problema al escalar.

🌐 Soporte Internacional

Idiomas Disponibles

✅ **Español** - Completamente traducido ✅ **Inglés** - Completamente traducido

**Cobertura:** 100% de la interfaz en ambos idiomas

Fácil de Agregar Más Idiomas

✅ El sistema está preparado para agregar más idiomas fácilmente ✅ Solo requiere crear archivo de traducciones nuevo

📦 Dependencias y Librerías

¿Están las librerías actualizadas?

- ✅ **Sí, en general todo está actualizado**
- Backend: Todas las librerías principales actualizadas
  - Frontend: Angular puede actualizarse a versión 18 (tiene 17)

¿Hay vulnerabilidades conocidas?

✅ **No se detectaron vulnerabilidades críticas** en dependencias

📊 Resumen Visual

Estado General por Área

Funcionalidad:	✅✅✅✅✅	95%
Organización:	✅✅✅✅⬜	80%
Seguridad:	✅✅⚠️⚠️⚠️	60%
Testing:	❌❌❌❌❌	0%
Performance:	✅✅✅⬜⬜	65%
Documentación:	✅✅⬜⬜⬜	40%
TOTAL:	✅✅✅⬜⬜	68%

Prioridades

Urgencia	Tema	Razón
----------	------	-------

● ALTA	Testing	Sin tests, cualquier cambio es arriesgado
● ALTA	CORS abierto	Riesgo de seguridad
● ALTA	Sync de BD	Puede causar pérdida de datos
● MEDIA	Límites de request	Riesgo de saturación
● MEDIA	Optimización de queries	Puede ser lento con muchos datos
● BAJA	Documentación	Facilita mantenimiento



## Conclusiones

### Resumen de 3 Puntos

- ✅ **La aplicación está funcionalmente completa**
  - Todas las features principales implementadas
  - Login múltiple, recetas, actividades, pagos
  - Tecnología moderna y actualizada
- ⚠️ **Faltan salvaguardas de calidad**
  - 0% de tests automatizados
  - Algunas configuraciones de seguridad abiertas
  - Sin optimizaciones para escalar
- ✅ **El código es mantenible**
  - Bien organizado y estructurado
  - Un equipo nuevo puede entenderlo
  - Base sólida para crecer

### ¿Está listo para producción?

**Funcionalmente:** ✅ Sí - La app funciona y tiene todas las features

**Calidad:** ⚠️ Con reservas - Necesita tests y ajustes de seguridad

**Escalabilidad:** ⚠️ Limitada - Necesita optimizaciones para muchos usuarios

### Analogía Final

El proyecto es como una **casa recién construida**:

- ✅ La estructura es sólida (arquitectura)
- ✅ Todas las habitaciones están (funcionalidades)
- ✅ Tiene puertas y ventanas (seguridad básica)
- ⚠️ Falta el sistema de alarma (tests)
- ⚠️ Algunas ventanas están abiertas (configuraciones de seguridad)
- ⚠️ No tiene inspector de calidad (testing automatizado)

**Puedes vivir en ella, pero es recomendable terminar estos detalles.**



## Glosario de Términos

Para facilitar la comprensión, aquí están los términos técnicos explicados:

Término	Explicación Simple
---------	--------------------



Backend	Servidor que maneja datos y lógica de negocio
Frontend	Interfaz que ve el usuario (la app móvil)
API	Forma en que la app se comunica con el servidor
CORS	Sistema de seguridad que controla quién puede acceder al servidor
JWT	Sistema de tokens para mantener sesión de usuario
Testing	Programas que verifican que el código funciona
TypeORM	Sistema que maneja la base de datos
Capacitor	Herramienta que permite acceder a funciones del teléfono
Índice (BD)	Como el índice de un libro, acelera búsquedas
Lazy Loading	Cargar solo lo necesario, no todo de golpe
Paginación	Mostrar resultados de 10 en 10, no todos juntos



## Información del Documento

**Preparado:** 18 de Octubre, 2025  
**Tipo:** Análisis de estado actual (sin proyecciones futuras)  
**Nivel:** Técnico-ejecutivo simplificado  
**Audiencia:** Directivos y stakeholders no técnicos  
**Versión:** 1.0

*Este documento describe únicamente el estado actual del proyecto, sin incluir recomendaciones de acción, estimaciones de tiempo o costos.*