

300 IOS Interview Questions

What is the LLVM?

LLVM es un proyecto general para muchos subproyectos. Todo lo cual da como resultado una infraestructura de compilación y una cadena de herramientas utilizada hoy en gran parte por los desarrolladores que utilizan lenguajes basados en C / C ++ y que está muy integrada en Xcode y su proceso de compilación.

<https://medium.com/@jyaunches/introduction-to-the-llvm-for-a-ios-engineer-8c00ed0f9ff0>

How could you setup Live Rendering?

Live Rendering, que nos permite personalizar nuestros objetos en tiempo de diseño, con el mismo aspecto en Interface Builder que en tiempo de ejecución

IBDesignable e IBInspectable permiten al desarrollador obtener una vista previa de la clase personalizada a través de xib.

https://www.rapidvaluesolutions.com/tech_blog/live-rendering-in-xcode6/

What is the difference between Synchronous & Asynchronous task?

Synchronous: espera hasta que la tarea se haya completado

Asynchronous: completa una tarea en segundo plano y puede notificarle cuando finalice

La clave está en la diferencia entre Synchronous y Asynchronous. Con el primero, sincrónico, implica que deben ocurrir dos cosas al mismo tiempo, o alternativamente, que una de ellas debe esperar hasta que la otra se ponga al día. Con este último, asíncrono, significa que las dos cosas no saben o no les importa cuando la otra está sucediendo.

Explain Compilation Conditions

Las declaraciones de control del compilador permiten que el

programa cambie aspectos del comportamiento del compilador. Swift tiene tres instrucciones de control del compilador:

un bloque de compilación condicional (**Conditional Compilation Block**)

permite que el código se compile condicionalmente según el valor de una o más condiciones de compilación.

Cada bloque de compilación condicional comienza con la directiva de compilación `#if` y termina con la directiva de compilación `#endif`.

```
#if compilation condition
    statements
#endif
```

una instrucción de control de línea (**line control statement**)

se usa para especificar un número de línea y un nombre de archivo que pueden ser diferentes del número de línea y el nombre de archivo del código fuente que se está compilando. Use una instrucción de control de línea para cambiar la ubicación del código fuente utilizada por Swift para fines de diagnóstico y depuración.

```
#sourceLocation(file: filename, line: line
    number)

#sourceLocation()
```

una instrucción de diagnóstico en tiempo de compilación (**compile-time diagnostic statement**)

hace que el compilador emita un error o una advertencia durante la compilación

```
#error("error message")
```

```
#warning("warning message")
```

<https://docs.swift.org/swift-book/ReferenceManual/Statements.html>

What is made up of NSError object?

Hay tres partes del objeto NSError: un dominio, un código de error y un diccionario de información del usuario. El dominio es una cadena que identifica de qué categorías de errores proviene este error.

What is Enum or Enumerations?

Una enumeración es un tipo de datos definido por el usuario que consiste en un conjunto de valores relacionados. La palabra clave enum se utiliza para definir el tipo de datos enumerados. Puede definir enumeraciones Swift para almacenar valores asociados de cualquier tipo dado, y los tipos de valores pueden ser diferentes para cada caso de la enumeración si es necesario. Las enumeraciones similares a estas se conocen como uniones discriminadas, uniones etiquetadas o variantes en otros lenguajes de programación.

What is the bounding box?

es el rectángulo más pequeño que encierra completamente todos los puntos de la ruta, incluidos los puntos de control para Bézier y las curvas cuadráticas.

Why don't we use strong for enum property in Objective-C?

What is @synthesize in Objective-C?

generará métodos getter y setter para su propiedad en tiempo de compilación.

What is @dynamic in Objective-C?

solo le dice al compilador que los métodos getter y setter no se implementan por la clase en sí, sino en otro lugar (como la superclase

o se proporcionarán en tiempo de ejecución).

Los usos de @dynamic son, por ejemplo, con subclases de NSObject(CoreData) o cuando desea crear una salida para una propiedad definida por una superclase que no se definió como una salida.

Why do we use synchronized?

La directiva @synchronized es una forma conveniente de crear bloqueos mutex (Exclusion mutua) sobre la marcha en código Objective-C. La directiva @synchronized hace lo que haría cualquier otro bloqueo mutex: evita que diferentes subprocesos adquieran el mismo bloqueo al mismo tiempo

What is the difference strong, weaks, read only and copy?

Una referencia **strong** significa que desea "poseer" el objeto al que hace referencia con una propiedad o variable. El compilador se encargará de que no se destruya ningún objeto que asigne a la propiedad. Solo una vez que establezca la propiedad nil, el objeto se destruirá (a menos que uno o más de los otros objetos también tengan una fuerte referencia a él).

Por el contrario, con una referencia **weak** , significa que no desea tener control sobre la vida útil del objeto. El objeto al que hace referencia débilmente solo vive porque al menos otro objeto tiene una fuerte referencia a él. Una vez que ya no sea así, el objeto se destruye y su propiedad débil se configurará automáticamente nil. Los casos de uso más frecuentes de referencias débiles en iOS son:

- delegar propiedades**, a las que a menudo se hace referencia débilmente para evitar retener ciclos, y
- subvistas** / controles de la vista principal de un controlador de vista porque esas vistas ya están fuertemente controladas por la vista principal.

read only como su nombre indica, el valor no se puede asignar al

objeto. Omite el método setter.

copy se necesita cuando el objeto es mutable. Use esto si necesita el valor del objeto tal como está en este momento, y no desea que ese valor refleje los cambios realizados por otros propietarios del objeto. Deberá liberar el objeto cuando haya terminado con él porque está reteniendo la copia.

<https://medium.com/bitmountn/attributes-of-property-nonatomic-retain-strong-weak-etc-b7ea93a0f772>

What is Dynamic Dispatch?

Dynamic Dispatch es el proceso de seleccionar qué implementación de una operación polimórfica es un método o una función para llamar en tiempo de ejecución. Esto significa que cuando queremos invocar nuestros métodos como método de objeto, pero Swift no admite el despacho dinámico directamente.

What's Code Coverage?

es una característica que le permite visualizar y medir la cantidad de código que ejercen las pruebas. Con la cobertura del código, puede determinar si sus pruebas están haciendo el trabajo que deseaba.

<https://medium.com/zendesk-engineering/code-coverage-and-xcode-6b2fb8756a51>

What's Completion Handler?

Un **completion handler** es básicamente solo una función que se pasa como parámetro de otra función.

La razón por la que nos gustaría hacer esto es porque queremos que se nos notifique cuando algo esté completo. Generalmente usamos funciones que devuelven un valor, pero esto solo es útil cuando estamos haciendo un trabajo sincrónico. Para el trabajo asíncrono, utilizamos controladores de finalización.

<https://programmingwithswift.com/understanding-completion-handlers-in-swift/>

How to Prioritize Usability in Design ?

¿Qué es la usabilidad?

La usabilidad es la facilidad de uso mediante la cual un usuario puede trabajar en un producto para lograr sus objetivos de una manera eficiente y efectiva. Es una medida de satisfacción del usuario cuando está usando un producto.

<https://blog.prototypr.io/usability-a-key-part-of-design-process-f0d33cf69a97>

What's the difference between the frame and the bounds?

Un objeto de vista rastrea su tamaño y ubicación utilizando sus propiedades de marco (**frame**), límites (**bounds**) y **centro**:

La propiedad de **frame** contiene el rectángulo de marco, que especifica el tamaño y la ubicación de la vista en el sistema de coordenadas de su supervista.

La propiedad de **bound** contiene el rectángulo de límites, que especifica el tamaño de la vista (y su origen de contenido) en el propio sistema de coordenadas local de la vista .

La propiedad del **centro** contiene el punto central conocido de la vista en el sistema de coordenadas de la supervista.

What is Responder Chain ?

El primer objeto en el ResponderChain se llama el primer respondedor. La cadena de respuesta es una serie de objetos de respuesta vinculados. Comienza con el primer respondedor y termina con el objeto de la aplicación. Si el primer respondedor no puede manejar un evento, reenvía el evento al siguiente respondedor en la cadena de respondedores

<https://medium.com/ios-os-x-development/understanding-cocoa-and-cocoa-touch-responder-chain-12fe558ebe97>

What is Regular expressions ?

una secuencia de caracteres que conforma un patrón de búsqueda

What is Operator Overloading ?

es la práctica de agregar nuevos operadores y modificar los existentes para hacer cosas diferentes

What is TVMLKit?

TVMLKit es un framework flexible basado en plantillas que es perfecto para crear hermosas aplicaciones de medios de transmisión para tvOS

What is Platform limitations of tvOS?

What is Functions?

son fragmentos de código independientes que realizan una tarea específica

What is ABI?

permite a los proveedores de sistemas operativos incorporar una biblioteca estándar de Swift y tiempo de ejecución en el sistema operativo que sea compatible con las aplicaciones creadas con Swift 5 o posterior.

ABI son las siglas de **Application Binary Interface**. En tiempo de ejecución, los binarios del programa Swift interactúan con otras bibliotecas a través de un ABI. Define muchos detalles de bajo nivel para entidades binarias, como cómo llamar a funciones, cómo se representan sus datos en la memoria, dónde están los metadatos y cómo acceder a ellos.

<https://medium.com/swift-india/swift-5-abi-stability-769ccb986d79>

Why is design pattern very important ?

nos ayudan a resolver problemas recurrentes de diseño. Tenga en cuenta que: los patrones de diseño no resuelven el problema por sí

mismos, nos ayudan a resolver el problema.

What is Singleton Pattern ?

El patrón singleton es un patrón de diseño.

El patrón singleton garantiza que solo se instancia una instancia de una clase.

<https://cocoacasts.com/what-is-a-singleton-and-how-to-create-one-in-swift>

What is Facade Design Pattern?

El patrón de diseño de Fachada proporciona una interfaz única para un subsistema complejo. En lugar de exponer al usuario a un conjunto de clases y sus API, solo expone una API unificada simple.

<https://www.raywenderlich.com/477-design-patterns-on-ios-using-swift-part-1-2>

What is Decorator Design Pattern?

El patrón Decorator agrega dinámicamente comportamientos y responsabilidades a un objeto sin modificar su código. Es una alternativa a la subclasificación donde modifica el comportamiento de una clase envolviéndola con otro objeto.

<https://www.raywenderlich.com/477-design-patterns-on-ios-using-swift-part-1-2>

What is Adapter Pattern?

Un adaptador permite que las clases con interfaces incompatibles trabajen juntas. Se envuelve alrededor de un objeto y expone una interfaz estándar para interactuar con ese objeto.

<https://www.raywenderlich.com/476-design-patterns-on-ios-using-swift-part-2-2>

What is Observer Pattern?

En el patrón Observador, un objeto notifica a otros objetos de cualquier cambio de estado. Los objetos involucrados no necesitan conocerse entre sí, lo que fomenta un diseño desacoplado. Este patrón se usa con mayor frecuencia para notificar a los objetos

interesados cuando una propiedad ha cambiado.

Cocoa implementa el patrón de observación de dos maneras: Notificaciones y Observación de valores clave (KVO) .

<https://www.raywenderlich.com/476-design-patterns-on-ios-using-swift-part-2-2>

What is Memento Pattern?

captura y externaliza el estado interno de un objeto. En otras palabras, guarda tus cosas en alguna parte. Más adelante, este estado externalizado puede restaurarse sin violar la encapsulación; es decir, los datos privados siguen siendo privados.

Explain MVC

El patrón de diseño de arquitectura MVC - Model View Controller es la arquitectura más común que se encuentra en las aplicaciones de iOS

Una aplicación que adopta el patrón MVC generalmente tiene 3 capas principales.

Model

Los modelos son representaciones de los datos de su aplicación. Por ejemplo, objetos como Usuarios o Publicaciones serían modelos de datos. Una clase o estructura de usuario podría contener información vinculada a un usuario específico como nombre de usuario, fecha de nacimiento, etc.

2. View

Las vistas son objetos con los que los usuarios de su aplicación pueden ver e interactuar. Los objetos de vista deben ser reutilizables y flexibles. Cosas como UILabels que pueden mostrar cualquier tipo de texto son vistas.

3. Controller

Los controladores median entre la Vista y el Modelo. Los controladores configuran vistas con los datos de los modelos y actualizan los modelos cuando los usuarios interactúan con las vistas.

Explain MVVM

Model-View-ViewModel (MVVM) es un patrón de diseño estructural que separa los objetos en tres grupos distintos:

Los modelos contienen datos de la aplicación. Suelen ser estructuras o clases simples.

Las vistas muestran elementos visuales y controles en la pantalla. Son típicamente subclasses de UIView.

Los modelos de vista transforman la información del modelo en valores que se pueden mostrar en una vista. Por lo general, son clases, por lo que se pueden pasar como referencias.

Use este patrón cuando necesite transformar modelos en otra representación para una vista

<https://www.raywenderlich.com/34-design-patterns-by-tutorials-mvvm>

How many different annotations available in Objective-C ?

Hay cuatro anotaciones

Null unspecified puentes a un Swift opcional envuelto implícitamente. Este es el valor predeterminado.

Non-null el valor no será nil; puentes a una referencia regular.

Nullable el valor puede ser nil; puentes a un opcional.

Non-null el valor nunca puede ser nil cuando se lee, pero puede configurarlo nil para reiniciarlo. Solo se aplica a propiedades.

Pointer	properties
_Null_unspecified	null_unspecified
_Nonnull	nonnull
_Nullable	nullable
N/A	null_resettable

<https://swiftunboxed.com/interop/objc-nullability-annotations/>

What is JSON/PLIST limits ?

Creamos objetos y luego los serializamos en el disco.

Es genial y casos de uso muy limitados.

Obviamente no podemos usar consultas complejas para filtrar nuestros resultados.

Es muy lento.

Cada vez que necesitamos algo, debemos serializarlo o deserializarlo.

No es seguro para subprocessos.

What is SQLite limits ?

significa tamaños o cantidades que no se pueden exceder.

- Maximum Database Size

- Número máximo de bases de datos adjuntas

- Número máximo de páginas en un archivo de base de datos

- Número máximo de tablas en un esquema

- Número máximo de filas en una tabla

- Número máximo de columnas

- Longitud máxima de una instrucción SQL

- Número máximo de tablas en una unión

- Longitud máxima de una cadena o BLOB

- Profundidad máxima de un árbol de expresión

- Profundidad máxima de recursividad de disparo

- Longitud máxima de un patrón LIKE o GLOB

<http://www.dbtalks.com/tutorials/learn-sqlite/what-are-the-limitations-of-sqlite>

What is Realm benefits ?

How many are there APIs for battery-efficient location tracking ?

Cambios de ubicación significativos: la ubicación se entrega aproximadamente cada 500 metros (generalmente hasta 1 km)

Monitoreo de región: realice un seguimiento de los eventos de entrada / salida desde regiones circulares con un radio igual a 100 m más. El monitoreo de la región es la API más precisa después del GPS.

Visitar eventos: supervisar el lugar Visitar eventos que entran o salen de un lugar (hogar / oficina).

What is the Swift main advantage ?

Legibilidad

La ventaja número uno para elegir Swift es posiblemente debido a su sintaxis limpia, que hace que sea más fácil de leer y escribir.

Mantenimiento

Swift elimina este requisito de dos archivos, combinando el encabezado Objective-C (.h) y los archivos de implementación (.m) en un solo archivo de código (.swift).

Plataforma más segura

Swift toma el código nulo y genera un error del compilador cuando los programadores escriben código incorrecto. Con Swift, puede compilar y corregir los errores mientras escribe el código

Swift funciona mejor y más rápido cuando se trata de pruebas de errores

Menos código y menos legado

Las clases no se dividen en dos partes; La interfaz y la implementación. Esto reduce la cantidad de archivos en el proyecto a la mitad, lo que hace que sea mucho más fácil de manejar.

Velocidad

Swift admite bibliotecas dinámicas

Las bibliotecas dinámicas son fragmentos de código ejecutables que se pueden vincular a una aplicación. Esta característica permite que las aplicaciones Swift actuales se vinculen con versiones más nuevas del lenguaje Swift a medida que evoluciona con el tiempo. Las bibliotecas dinámicas en Swift se cargan directamente en la memoria, lo que reduce el tamaño inicial de la aplicación y, en última instancia, aumenta el rendimiento de la aplicación.

'Playgrounds' fomenta la codificación interactiva

fuentes abiertas

Swift de código abierto significa que Apple podrá recibir comentarios de la comunidad para realizar mejoras de manera consistente a medida que los desarrolladores independientes contribuyan al éxito del lenguaje. Swift no solo ha despegado con éxito porque está bien estructurado y diseñado, sino también porque muchos

desarrolladores lo han respaldado.

Explain generics in Swift ?

El código genérico le permite escribir funciones y tipos flexibles y reutilizables que pueden funcionar con cualquier tipo, sujeto a los requisitos que defina. Puede escribir código que evite la duplicación y exprese su intención de manera clara y abstracta.

<https://www.appcoda.com/swift-generics/>

Explain lazy in Swift ?

El valor inicial de las lazy propiedades almacenadas **se calcula solo cuando se llama a la propiedad por primera vez.**

Las propiedades diferidas son útiles cuando el valor inicial de una propiedad depende de factores externos cuyos valores no se conocen hasta que se completa la inicialización de una instancia. También son útiles cuando el valor inicial de una propiedad requiere una configuración compleja o computacionalmente costosa que no debe realizarse a menos que sea necesario.

Reglas Lazy:

- No puedes usar lazy con let.

- No puedes usarlo con computed properties. Porque, una propiedad calculada devuelve el valor cada vez que intentamos acceder después de ejecutar el código dentro del bloque de cálculo.

- Puede usar lazy solo con miembros de struct y class.

- Las variables lazy no se inicializan atómicamente y, por lo tanto, no es seguro para subprocesos.

Explain what is defer ?

palabra clave que proporciona un bloque de código que se ejecutará en el caso cuando la ejecución salga del alcance actual

How to pass a variable as a reference ?

En Swift, las instancias de clases se pasan por referencia

Las instancias de estructuras y enumeraciones se pasan por

valor . Cada instancia de una estructura o enumeración tiene su propia copia única de datos

<https://cocoacasts.com/value-types-and-reference-types-in-swift>

How to pass data between view controllers?

Mediante el uso de una propiedad de instancia ($A \rightarrow B$)

Mediante el uso de segues (para Storyboards)

Mediante el uso de propiedades y funciones de instancia ($A \leftarrow B$)

Mediante el uso del patrón de **delegación**

Mediante el uso de un controlador de cierre o finalización

Mediante el uso **NotificationCenter** y **el patrón Observador**

<https://learnappmaking.com/pass-data-between-view-controllers-swift-how-to/>

What is Concurrency ?

La concurrencia significa que una aplicación está progresando en más de una tarea al mismo tiempo (simultáneamente)

<https://medium.com/swift-india/parallel-programming-with-swift-part-1-4-df7caac564ae>

Grand Central Dispatch (GCD)

Grand Central Dispatch o GCD es una API de bajo nivel para administrar operaciones concurrentes.

Grand Central Dispatch (GCD) es una API de bajo nivel para administrar operaciones de forma asíncrona o sincrónica. GCD se puede usar para administrar tareas pesadas en segundo plano para que podamos mejorar la capacidad de respuesta de nuestra aplicación.

Grand Central Dispatch (GCD) es una API basada en cola que permite ejecutar cierres en grupos de trabajadores en el orden Primero en entrar, primero en salir . El orden de finalización dependerá de la duración de cada trabajo.

Readers-Writers

El problema de lectores-escritores ocurre cuando múltiples hilos

comparten un recurso (variable o una estructura de datos en la memoria), y algunos hilos pueden escribir, otros pueden leer. Cuando se leen múltiples hilos del recurso compartido simultáneamente, no puede suceder nada malo. Pero cuando un hilo escribe, otros deben esperar, de lo contrario terminan con datos corruptos. Es decir, el problema de lectores-escriitores permite la lectura concurrente pero la escritura es serial.

NSOperation — NSOperationQueue — NSBlockOperation
KVC — KVO

Please explain Swift's pattern matching techniques

un patrón representa la estructura de un valor único o compuesto. hay ocho diferentes tipos de patrones que podemos utilizar en conjunción las sentencia **if** , la sentencia **switch**, la cláusula **catch** de sentencia **do** o en la condición de la sentencias **if**, **while**, **guard** o **for - in**

- Identifier Patterns
- Wildcard Patterns
- Value Binding Patterns
- Tuple Patterns
- Enumeration Case Patterns
- Optional Patterns
- Type Casting Patterns
- Expressions Patterns

<https://andybargh.com/pattern-matching-in-swift/>

Explain Guard statement

es una declaración condicional que requiere una expresión para evaluar true para que la ejecución continúe. Si la expresión es false, la else cláusula obligatoria se ejecuta en su lugar.

<https://nshipster.com/guard-and-defer/>

Please explain Method Swizzling in Swift

Swizzling es el acto de cambiar la funcionalidad de un método al reemplazar la implementación de ese método con otro, generalmente en tiempo de ejecución.

Hay muchas razones diferentes por las que uno podría desear utilizar swizzling: introspección, anulación del comportamiento predeterminado o tal vez incluso la carga dinámica del método

<https://medium.com/@abhimuralidharan/method-swizzling-in-ios-swift-1f38edaf984f>

What is the difference Non-Escaping and Escaping Closures?

<https://buildingyts.com/escaping-non-escaping-closures-memory-management-and-you-cb936c60a9d1>

Explain [Weak self] and [unowned self] ?

Unowned and Weak funciona de la misma manera para superar el problema del ciclo de retención de la administración de memoria. Ambos no mantendrán un fuerte control sobre la instancia a la que se refiere, detrás de la puerta no aumentarán el conteo del objeto que se refiere.

Unowned no puede ser nula

Weak si puede ser nula

<https://medium.com/@kiran.jasvane/difference-between-unowned-self-and-weak-self-in-swift-310c14961953>

What is ARC ?

El Conteo automático de referencia (ARC) es el sistema de memoria de seguimiento de Swift que está utilizando. Cuando crea un objeto a partir de una clase, Swift recuerda que se hace referencia a esa instancia con precisión una vez. Si luego apunta otra variable a ese objeto, Swift incrementará el recuento de referencia a 2, porque dos variables apuntan al mismo objeto. Si ahora destruye la primera variable (tal vez estaba dentro de una función y esa función terminó), Swift vuelve a reducir el recuento de referencias a 1.

Todo esto importa porque mientras el recuento de referencias sea mayor que 1, el objeto permanecerá vivo. Pero cuando la variable final que hace referencia a ese objeto desaparece, Swift llevará la cuenta de referencia a cero. Como ninguna variable existente apunta al objeto, se puede liberar su RAM

Retain Cycle (ciclo de retención)

Podría ser un escenario en el que dos instancias de clase mantienen una fuerte referencia entre sí y no hay forma de que el sistema las desasigne.

Esto significa que el Recuento de retención de ambas clases de la clase nunca bajaría a 0. Esto se conoce como Strong Reference Cycle.

Escenarios comunes para ciclos de retención:

Delegados

Closure

Explain #keyPath() ?

son referencias a una propiedad de un tipo. Son útiles cuando desea realizar operaciones dependiendo de esa propiedad, más bien del valor detrás de ella. Esto los hace útiles especialmente para manejar colecciones y observar valores clave.

<https://medium.com/joshtastic-blog/swift-keypaths-ea5a1ec880e9>

What is iOS 11 SDK Features for Developers?

What makes React Native special for iOS?

What is NSFetchRequest? Explain NSPersistentContainer

Explain NSFetchedResultsController

What is the three major debugging improvements in Xcode 8?

What is the Test Driven Development of three simple rules?

1. No está permitido escribir ningún código de producción a menos que sea para aprobar una prueba de unidad que falla.
2. No se le permite escribir más de una prueba unitaria de la que

es suficiente para fallar; y las fallas de compilación son fallas.

3. No se le permite escribir más código de producción del que sea suficiente para pasar la prueba de la unidad que falla.

Please explain final keyword into the class?

A veces quieres evitar que otros desarrolladores creen su propia clase basada en la tuya.

cuando declaras que una clase es final, ninguna otra clase puede heredarla. Esto significa que no pueden anular sus métodos para cambiar su comportamiento; necesitan usar su clase de la manera en que fue escrita.

What does Yak Shaving mean?

Es un término de programación que se refiere a una serie de tareas que deben realizarse antes de que un proyecto pueda avanzar a su próximo hito

Es una jerga de programación para la serie aparentemente interminable de pequeñas tareas que deben completarse antes de que el siguiente paso en un proyecto pueda avanzar

What is the difference open & public access level?

public: Las clases y los miembros de la clase solo pueden subclasificarse y overridden dentro del módulo de definición (destino) .

open: Las clases y los miembros de la clase pueden subclasificarse y overridden tanto dentro como fuera del módulo de definición (destino).

<https://medium.com/@abhimuralidharan/swift-3-0-1-access-control-9e71d641a56c>

What is the difference fileprivate, private and public private(set) access level ?

fileprivate restringe el uso de una entidad a su propio archivo fuente de definición.

private restringe el uso de una entidad a la declaración adjunta y a las

extensiones de esa declaración que están en el mismo archivo.

<https://medium.com/@chetan15aga/ios-interview-questions-part-3-swift-4b00ce352122>

What is Internal access ?

permite que las entidades se usen dentro de cualquier archivo fuente desde su módulo de definición, pero no en ningún archivo fuente fuera de ese módulo

<https://medium.com/@abhimuralidharan/swift-3-0-1-access-control-9e71d641a56c>

What is the difference between BDD and TDD?

En **TDD** (Test Driven Development), la prueba se escribe para verificar la implementación de la funcionalidad, pero a medida que el código evoluciona, las pruebas pueden dar resultados falsos.

BDD (Behavior Driven Development) también es un enfoque de prueba primero, pero difiere al probar el comportamiento real del sistema desde la perspectiva de los usuarios finales.

Please explain "Arrange-Act-Assert"

Es un patrón de prueba para describir las fases naturales de la mayoría de las pruebas de software.

Organizar describe cualquier configuración necesaria

Act describe el comportamiento del sujeto que está bajo prueba (y generalmente solo describe una sola línea necesaria para invocar ese comportamiento)

Assert describe la verificación de que el comportamiento del sujeto tuvo el efecto deseado evaluando su valor de retorno o midiendo un efecto secundario (con un espía o simulacro)

Muchas personas han identificado que "**Given**", "**When**" y "**Then**" son descripciones más naturales en inglés de las tres frases, y como resultado, muchas herramientas BDD nombraron sus palabras clave de prueba en consecuencia.

What is the benefit of writing tests in iOS apps?

Hace que el proceso sea mas agil

Calidad del código

Encuentre los errores de software temprano

Facilita los cambios y simplifican la integración

Proporcionan documentación

Simplifican el proceso de depuración

Escribir pruebas obliga a pensar en el diseño

Encontrar los errores temprano en las pruebas ayuda a reducir el costo.

<https://blog.usejournal.com/the-ios-testing-manifesto-e1bc821cc4c3>

<https://apiumhub.com/tech-blog-barcelona/top-benefits-of-unit-testing/>

What is five essential practical guidelines to improve your typographic quality of mobile product designs?

1. Comience eligiendo el tipo de letra de su cuerpo.
2. Intente evitar mezclar tipos de letra.
3. Observe la longitud de su línea.
4. Equilibre la altura de la línea y el tamaño del punto.
5. Use apóstrofes y guiones adecuados.

Explain Forced Unwrapping

Es la acción de extraer el valor contenido dentro de un Optional. Esta operación es peligrosa porque le está diciendo al compilador: estoy seguro de que este valor opcional contiene un valor real

<https://stackoverflow.com/questions/38817494/difference-between-force-unwrapping-optionals-and-implicitly-unwrapped-optionals>

How to educate app with Context?

What is bitcode ?

Bitcode es una representación intermedia de un programa compilado. Las aplicaciones que cargue en iTunes Connect que contengan código de bits se compilarán y vincularán en la App

Store. La inclusión de código de bits permitirá a Apple volver a optimizar el binario de su aplicación en el futuro sin la necesidad de enviar una nueva versión de su aplicación a la tienda.

<https://www.guardsquare.com/es/blog/it-safe-enable-bitcode>

Explain Swift Standard Library Protocol ?

define una capa base de funcionalidad para escribir programas Swift, que incluye:

- Tipos de datos fundamentales, tales como `Int`, `Double`, y `String`

- Estructuras de datos común, como `Array`, `Dictionary` y `Set`

- Funciones globales como `print(_:separator:terminator:)` `abs(_:)`

- Protocolos, como `Collection` y `Equatable`, que describen abstracciones comunes.

- Protocolos, como `Comparable` y `ExpressibleByStringLiteral`, que utiliza para personalizar operaciones que están disponibles para todos los tipos.

- `CustomDebugStringConvertible` `CustomReflectable`

- Protocolos, como `OptionalProtocol` que utiliza para proporcionar implementaciones que de otro modo requerirían código repetitivo. `OptionSet`

What is the difference SVN and Git ?

SVN se basa en un sistema centralizado para la gestión de versiones. Es un repositorio central donde se generan copias de trabajo y se requiere una conexión de red para acceder.

Git se basa en un sistema distribuido para la gestión de versiones. Tendrás un repositorio local en el que puedes trabajar, con una conexión de red solo necesaria para sincronizar.

What is the difference CollectionViews & TableViews?

TableViews muestra una lista de elementos, en una sola columna, de forma vertical y limitada al desplazamiento vertical solamente.

CollectionViews también muestra una lista de elementos, sin embargo, pueden tener múltiples columnas y filas.

What is Alamofire doing?

Alamofire es una forma elegante de interactuar con las solicitudes de red HTTP. Se basa en el sistema de carga de URL de Apple proporcionado por el marco de la Fundación. En el núcleo del sistema se encuentran URLSession y las subclasses URLSessionTask. Alamofire envuelve estas API, y muchas otras, en una interfaz más fácil de usar y proporciona una variedad de funcionalidades necesarias para el desarrollo moderno de aplicaciones utilizando redes HTTP.

REST, HTTP, JSON — What's that?

REST es la abreviatura de REPRESENTATIONAL STATE TRANSFER.

una técnica de arquitectura de software usada para construir APIs que permitan comunicar a nuestro servidor con sus clientes usando el protocolo HTTP mediante URIs lo suficientemente inteligentes para poder satisfacer la necesidad del cliente.

HTTP es un protocolo de acceso para las páginas web a través de Internet. HTTP son las siglas para Hypertext Transfer Protocol que se traduce al español como el “protocolo de transferencia de hipertextos”.

JSON es el acrónimo para JavaScript Object Notation, y aunque su nombre lo diga, no es necesariamente parte de JavaScript, de hecho es un estándar basado en texto plano para el intercambio de información, por lo que se usa en muchos sistemas que requieren mostrar o enviar información para ser interpretada por otros sistemas,

JSON es un formato de texto para la serialización de datos estructurados.

JSON puede representar cuatro tipos primitivos(cadenas, números, booleanos, valores nulos) y dos tipos estructurados(objetos y arreglos).

What problems does delegation solve?

Evite el acoplamiento apretado de objetos.

Modifique el comportamiento y la apariencia sin la necesidad de subclase de objetos.

Permita que las tareas se manejen a cualquier objeto arbitrario.

What is the major purposes of Frameworks?

Los Frameworks sirven para el mismo propósito que las bibliotecas compartidas estáticas y dinámicas, es decir, proporcionan una biblioteca de rutinas a las que puede llamar una aplicación para realizar una tarea específica.

Explain Swift Package Manager

es una herramienta para administrar la distribución del código Swift. Está integrado con el sistema de compilación Swift para automatizar el proceso de descarga, compilación y vinculación de dependencias.

<https://medium.com/developerinsider/create-a-library-using-swift-package-manager-stepwise-tutorial-181604ec8a7>

What is the difference between a delegate and an NSNotification?

Un delegado usa protocolos y crea una asociación entre las dos clases. Uno de los otros beneficios de los delegados es que puede devolver algo a la clase propietaria.

Las notificaciones, por otro lado, están más orientadas a la comunicación punto a multipunto.

Explain SiriKit Limitations

Why do we use a delegate pattern to be notified of the text field's events?

How is an inout parameter different from a regular parameter?

Explain View Controller Lifecycle events order?

What is the difference between LLVM and Clang?

What is Class ?

es un modelo o prototipo para crear objetos. Básicamente, una clase

consta de propiedades y métodos.

What is Object?

Es la instancia de una clase

What is interface?

Una interfaz es una colección de métodos abstractos.

Un método abstracto es un método que se declara pero no contiene implementación

When and why do we use an object as opposed to a struct?

What is UINavigationController?

proporciona una forma de diseñar una serie de vistas horizontal o verticalmente. Podemos definir cómo las vistas contenidas se ajustan al espacio disponible

What are the states of an iOS App?

What are the most important application delegate methods a developer should handle?

What is the difference between property and instance variable?

How can we add UIKit for Swift Package Manager? Explain the difference between SDK and Framework

What is Downcasting?

Explain Labeled Statements

What is Nil Coalescing & Ternary Operator?

Nil Coalescing (a ?? b) devuelve un opcional a si contiene un valor, o devuelve un valor predeterminado b si a es nil. La expresión siempre es de tipo opcional. La expresión b debe coincidir con el tipo que está almacenado dentro a

Ternary Operator es un operador especial con tres partes, que toma la forma question ? answer1 : answer2. Es un atajo para evaluar una de dos expresiones en función de si question es verdadero o falso. Si question es verdadero, evalúa answer1 y devuelve su

valor; de lo contrario, evalúa answer2y devuelve su valor.

- 1 What kind of JSONSerialization have ReadingOptions?
- 2 How can you prevent a user from doing said action more than once on their device?
- 3 What is DispatchGroup?
- 4 What is RGR (Red — Green — Refactor)?
- 5 Where do we use Dependency Injection ?
- 6 Please explain types of notifications.
- 7 When is a good time for dependency injection in our projects?
- 8 What kind of order functions can we use on collection types ?
- 9 What allows you to combine your commits ?

What is the difference ANY and ANYOBJECT ?

Any puede representar una instancia de cualquier tipo, incluidos los tipos de función y los tipos opcionales.

AnyObject puede representar una instancia de cualquier tipo de clase .

- 10 Please explain SOAP and REST Basics differences ?

What is your favorite Visualize Chart library?

- 11 What is the difference Filter and Map Function?
- 12 What is CoreData ?

- 13 Could you explain Associated type ?
- 14 Which git command saves your code without making a commit ?
- 15 Explain Priority Inversion and Priority Inheritance.
- 16 What is Heritable?

When do you use optional chaining vs. if let or guard?

optional chaining es un proceso para consultar y llamar a propiedades, métodos y subíndices en un opcional que podría ser actualmente nil. Si el opcional contiene un valor, la llamada de propiedad, método o subíndice tiene éxito; y si el opcional es nil, entonces vuelve nil. Se pueden encadenar varias consultas juntas, y la cadena completa falla correctamente si algún enlace de la cadena lo es nil.

<https://andybargh.com/optional-chaining/>

- 17 How many different ways to pass data in Swift?
- 18 How do you follow up clean code for this project?
- 19 Explain to using Class and Inheritance benefits
- 20 What's the difference optional between nil and . None?
- 21 What is GraphQL?
- 22 Explain Common features of Protocols & superclasses

What is Continuous Integration?

La **integración continua** (CI) es una práctica de desarrollo que requiere que los desarrolladores integren el código en un repositorio compartido varias veces al día. Luego, cada check-in es verificado por una compilación automatizada, lo que permite a los equipos detectar problemas temprano. Hay muchas herramientas de integración continua disponibles como: servidor Xcode, Jenkins, Travis, Fastlane, etc.

- 23 What is the difference Delegates and Callbacks?
- 24 Explain Linked List
- 25 Do you know Back End development?
- 26 Explain AutoLayout

- 27 What is the disadvantage to hard-coding log statements ?

What is Pointer ?

Un puntero es una variable cuyo valor es la dirección de otra variable, es decir, la dirección directa de la ubicación de la memoria.

Explain Core ML Pros and Cons

What is pair programming?

pair programming es una práctica ágil de desarrollo de software en la que dos programadores se unen en una estación de trabajo para maximizar la eficiencia. Con la programación de pares, uno de los dos programadores (el conductor) escribe el código mientras el otro observa y revisa (el observador). Los dos programadores cambian

roles con frecuencia

30 Explain blocks

What is Keychain ?

Keychain es una API para la persistencia de datos de forma segura en la aplicación iOS.

What is the biggest changes in UserNotifications ?

Explain the difference between atomic and nonatomic synthesized properties

Why do we use availability attributes ?

How could we get device token ?

What is Encapsulation ?

What is big-o notation ?

What Is Dependency Management?

What is UML Class Diagrams?

Explain throw

What is Protocol Extensions?

What is three triggers for a local notification ?

(**Location, Calendar, and Time Interval**) Ubicación, calendario e intervalo de tiempo . Se activa una notificación de ubicación cuando el GPS de su teléfono está en una ubicación o región geográfica. El activador de calendario se basa en datos de calendario divididos en componentes de fecha. El intervalo de tiempo es un recuento de segundos hasta que el temporizador se apaga.

Explain Selectors in ObjC

What is Remote Notifications attachment's limits ?

Explain Data Structures

Explain CodingKey Protocol

Explain IGListKit

What is URLSession?

How do we download images?

How does TestFlight make a difference?

What is SnapKit or Masonry make a difference from Auto layout?

Explain IteratorProtocol

Explain differences between WKWebView and UIWebView

Explain XLIFF Pros and Cons

What is difference Layout Margins and Directional Layout Margins ?

What is Sequence protocol ?

What is OAuth ?

What is offset ?

Explain rethrows keyword

Explain @objc inference

What is Safe area ?

Explain Viper Architecture

Explain Content offset

Explain Queues

Explain @objcMembers inference

What's new with Xcode Server ?

Explain type inference

si no especifica el tipo de valor que necesita, Swift usa la inferencia de tipos para calcular el tipo apropiado. La inferencia de tipos permite que un compilador deduzca el tipo de una expresión particular automáticamente cuando compila su código , simplemente examinando los valores que proporciona.

What is the difference between Array vs NSArray ?

What is NSLayoutAnchor ?

What is the mutating keyword ?

What is snapshot testing ?

What is circular dependencies ?

What's the difference between MKAnnotation and MKPointAnnotation?

What is DTO ?

Explain Main Thread Checker

What is the difference Stack and Heap ?

Stack son de acceso rápido y no tienen que desasignar explícitamente las variables. El espacio es administrado eficientemente por la CPU y la memoria no se fragmentará. Hay un límite en el tamaño de la pila (depende del sistema operativo). Por lo general, las variables se almacenan y no se pueden redimensionar.

Heap, se puede acceder a la variable de forma global, pero tiene un acceso lento. No tiene límite de memoria. No se garantiza un uso eficiente del espacio, la memoria puede fragmentarse con el tiempo a medida que se asignan bloques de memoria y luego se liberan. Necesitamos gestionar la memoria.

Explain VIP (Clean-Swift) Architecture

Explain UIBezierPath

Explain Dependency Injection Container

Explain ObjectMapper for parsing JSON data

Explain CAShapeLayer

Explain Get Request Steps

Explain coordinate systems in views

Explain Encoding, Decoding and Serialization, Deserialization

What is the purpose of using IBAction and IBOutlet ?

Explain Alamofire Benefits

Explain Semaphore in iOS

What is LLDB?

Explain Tuples

Explain the difference between Generics and AnyObject in Swift

Explain Dependency Inversion Principle

What is Smoke Testing?

Explain how does UIEdgeInsetsMake work?

What is the meaning of id ?

What excites or interests you about making software?

Which version control systems are you familiar with?

Do you have experience working with projects on GitHub?

Do you contribute to any iOS open source projects on GitHub or a

similar site?

Can you describe your workflow when you work on creating an iOS app?

Are you familiar with CocoaPods? Can you explain what they are and how they work?

In general, explain software licensing and how this applies to work we do.

Describe your general testing practices when building an iOS app.

How can apps support other languages, date formats and currencies?

What is Instruments and how is it useful?

Explain Handoff, and in general, how it allows iOS and Mac/web apps to communicate?

What technologies/services are contained inside of iCloud?

What is an iOS extension? Can you list a few examples of popular/common extensions?

What is HealthKit?

HealthKit es un framework en iOS. Almacena datos de salud y estado físico en una ubicación central. Toma datos de múltiples fuentes, que podrían ser dispositivos diferentes. Permite a los usuarios controlar el acceso a sus datos y mantiene la privacidad de los datos del usuario. Los datos se sincronizan entre su teléfono y su reloj.

What is HomeKit?

What is Apple Pay? Could you describe a way we could use it in our applications?

Explain application sandboxing.

What is VoiceOver? Explain some of the accessibility features included in iOS and how developers can utilize them.

How does application multitasking work on iOS?

What features does Game Center offer for iOS games?

What are iBeacons?

What is Cocoa/Cocoa Touch?

1) Cocoa, que incluye los framework Foundation y AppKit, se usa para desarrollar aplicaciones que se ejecutan en OS X.

2) Cocoa Touch, que incluye los framework Foundation y UIKit, se utiliza para desarrollar aplicaciones que se ejecutan en iOS.

Explain in general, what Core Audio, Core Data, and Core Location are, and how they help iOS apps.

Describe the role of SpriteKit and SceneKit.

What is Metal?

What is the responder chain? How does it work?

El primer objeto en el ResponderChain se llama el primer respondedor. La cadena de respuesta es una serie de objetos de respuesta vinculados. Comienza con el primer respondedor y termina con el objeto de la aplicación. Si el primer respondedor no puede manejar un evento, reenvía el evento al siguiente respondedor en la cadena de respondedores

What are the different actions that buttons and other controls can respond to?

What is the role of the application delegate?

Explain NSUserDefaults. How would you serialize an array to disk?

How would you store user's credentials?

Explain Keychain Services.

Why are caching and compression important on mobile devices?

Explain ~/Documents, ~/Library, ~/tmp. What directory is ~ on iOS?

How does AirPlay work? How would you use it (programmatically) to enhance the utility and presentation of an app?

Give a brief overview of what sensors, IO and communication methods (Wifi, telephony, etc) are available on iOS. How can you make use of these?

What are the hardware performance differences among the iPad 2 / iPad mini 1-3, iPad Retina, iPad Air 2, iPhone 5, 5s, 6, and 6+.

What do these constraints mean for performance intensive apps?

What does Cocoa Touch include and not include?

Why do Cocoa Touch classes start with two capital letters?

What is Swift, what is Objective-C and how do they relate and compare?

Explain why optionals are useful in Swift.

Explain NSError and how it works (or doesn't) in Swift.

How doesinstancetype work and how is it useful?

When is let appropriate in Swift? var?

Why and where is the map function useful.

How do you track down bugs? What are your tools of choice?

You found a bug in Cocoa. What do you do?

There is a regression in a new distributed version of our app causing crashes. How do you mitigate it? How will you prevent new bugs from reaching customers?

How are Objective-C classes implemented? Describe how the Objective-C runtime is implemented.

What security does iOS offer to protect customers and privileged information?

Our app downloads data and displays it immediately. In accordance with MVC where is the best place to perform the download?

How does MVC influence the design of a codebase?

What methods are part of the controller life-cycle? The view life-cycle

What design patterns does iOS make use of? What design patterns do you use in your codebase?

What queues does iOS provide and how can you best utilize them?

Give a brief description of how UIScrollView is implemented. How does it operate with gesture recognizers, multiple touches and the run loops?

What API would you add or improve on iOS?

What is the screen resolution of the iPhone 5, 6, 6+. and iPad Air 2?

What units is the resolution measured in?

Explain the purpose of Interface Builder, what is a NIB file?

What image filetype should iOS UI assets be saved in?

Describe some differences between a Storyboard and a standard NIB file.

What is the device status bar? How tall is it in points? Is it opaque or transparent? What does it do during a phone call or navigation?

What is a navigation bar? Can you show me an Apple app on your phone that uses a navigation bar?

What is a tab bar? What is a toolbar? Compare and contrast them.

What is a table view? What is a collection view?

Describe when a popover is most appropriate.

What is a split-view controller?

What sort of content would be appropriate to place in a picker view?

When are a label, text field and text view appropriate?

What does a segmented control do?

What is a modal view?

What kind of notifications does iOS offer?

What is an iOS app icon? Describe it as best as you can.

What is the smallest size an app icon could be? What's the largest size it could be?

Can an app icon contain any transparency?

How does a Newsstand icon differ from a regular app icon?

Explain a launch image.

Describe the purpose of Auto Layout, and in general, how it works.

Describe the role of animation in design of software.

Describe the role of interactivity and feedback when designing software.

What are some differences to take into account when building an iPhone app vs an iPad app?

Describe the importance and role of prototyping when working on an app design.

How do In-App Purchases work? What can be purchased with IAP?

Have you ever submitted an app to the App Store? Can you explain the general process?

What is iTunes Connect?

What is a provisioning profile?

What is an App ID?

What are the differences between Development and Production iOS signing certificates?

How is TestFlight used? How were UUIDs used in ad-hoc app distribution?

When do purchase receipts need to be verified?

What is required to display iAds?

What's a cool thing you've coded recently? What's something you've built that you're proud of?

What are some things you like about the developer tools you use?

Who are some of your favourite independent Mac or iOS developers?

Do you have any pet projects? What kind?

What would you change about Xcode?

What is your favourite iOS API?

What are your favourite ways to investigate new technology?

Why are dictionaries called dictionaries? Why not hash tables or hash maps?

Notificaciones

KVO: el propósito principal de este mecanismo es observar los cambios de una variable de objeto o una propiedad.

Es más fácil para usted entender cuando trata de responder a la pregunta: "¿Cómo puede saber que se modifica alguna propiedad o variable de una clase?".

En iOS, tenemos una forma sencilla de lograr esto utilizando KVO.

Un objeto primero se registra para observar una propiedad de otro objeto. Luego, cada vez que la propiedad de ese objeto cambia su valor,

se notifica automáticamente al observador con los cambios actualizados.

KVO no es bueno en Swift, ya que tiene que depender del tiempo de ejecución de Objective-C:

Debe usar @objc palabras clave con clases que hereden NSObject, luego marque cada una de sus propiedades @objc dynamic.

NotificationCenter

NotificationCenter es una clase de tipo Singleton que nos permite gestionar notificaciones que son enviadas por una clase, y recibidas por otra cualquiera que quiera suscribirse a la misma.

NotificationCenter es particularmente útil cuando hay varias instancias de clase o estructura que deben tomar medidas basadas en algo que sucede en otra parte de su aplicación.

Para este tipo de escenario, NotificationCenter puede ser una

gran herramienta para manejar al desarrollar aplicaciones en Swift para iOS.

Cuándo usar ?

NotificationCenter debe usarse cuando desee transmitir a través de todo su programa que hay cambios en toda la aplicación.

Esto le permite responder en múltiples clases que están observando para la misma notificación.

La mayor diferencia entre KVO y NotificationCenter es que KVO rastrea cambios específicos en un objeto,

mientras que NotificationCenter se usa para rastrear eventos genéricos, como cuando se presiona un botón para publicar una acción.

Si bien KVO le proporcionará la información que se ha cambiado, NotificationCenter solo le informa que ocurrió el evento.

Depende de usted cómo desea responder a ese evento.

Ahora puede pasar datos adicionales junto con la Notificación, pero tiene que configurarlo manualmente, en lugar de que venga automáticamente con KVO.

Pros y contras

ventajas: cuando usamos NotificationCenter, ni el remitente ni el destinatario de las notificaciones deben conocerse, también podemos especificar el método exacto para recibir la notificación.

contras: tenemos que publicar notificaciones para notificar a los observadores.

pros : KVO notifica automáticamente cualquier cambio en el observador,

contras : En KVO tenemos que subclase NSObject y anular observeValueForKeyPath: método, tener que usar dinámico modificador de la propiedad de observación para permitir KVO.

UserNotifications

En el desarrollo de iOS, Apple proporciona un marco

denominado UserNotifications

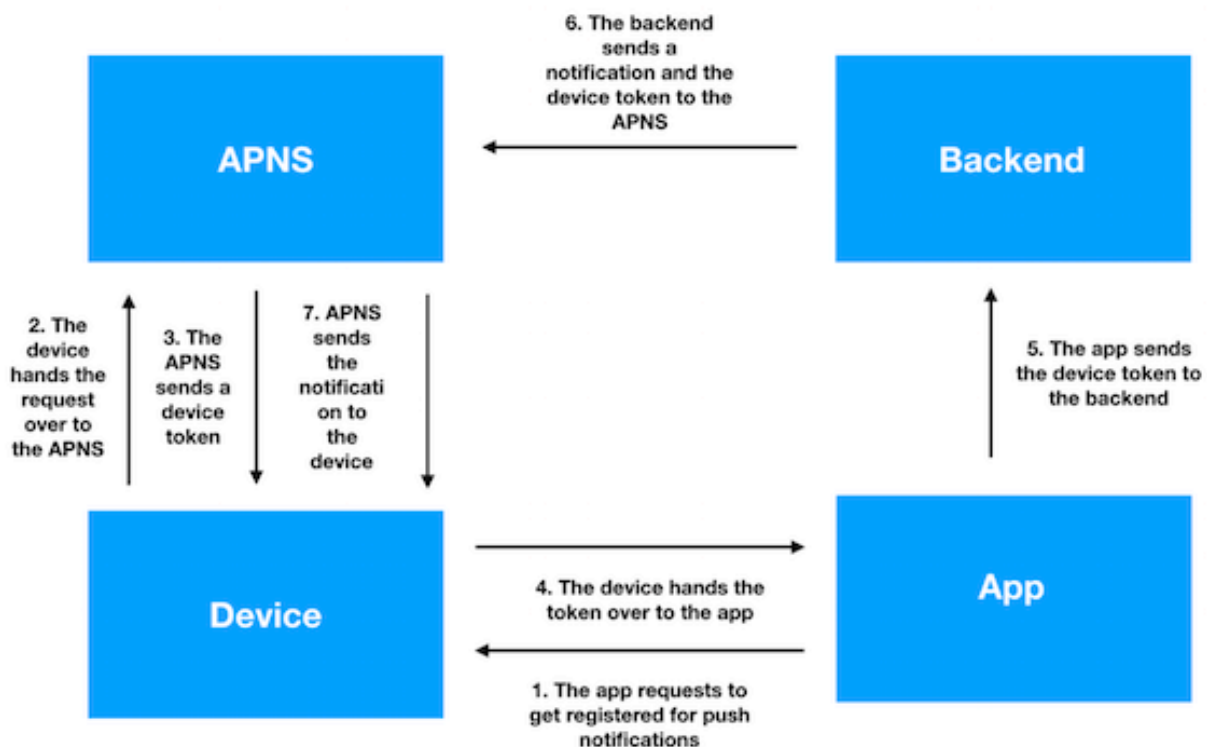
Push Notification

¿Cómo funcionan las notificaciones push?

En términos generales, hay cuatro participantes en todo el flujo de trabajo de notificaciones push:

la aplicación,
el dispositivo en el que se ejecuta la aplicación,
un servidor y el Servicio de notificaciones push de Apple (APNS).

Eche un vistazo al flujo de trabajo general:



1. Primero, la aplicación solicita registrarse para notificaciones remotas.
2. El dispositivo entrega la solicitud a la APNS.
3. Cuando el registro fue exitoso, el APNS envía un token al dispositivo.

4. El dispositivo entrega el token a la aplicación.
5. La aplicación envía el token del dispositivo al backend (o algún tipo de servicio en la nube), para que el backend sepa cómo puede identificar el dispositivo.
6. Cuando el backend desea enviar una notificación a un dispositivo, envía el mensaje y el token del dispositivo correspondiente al APNS.
7. El APNS envía la notificación al dispositivo.

Gestore

UIGestureRecognizer (Tap, pinch, pan, rotation)

Persistencia local

UserDefault

keyChain

Controles

UITableView

UICollectionView

UITextField

UIAlertViewController

UIPageController

WKWebView

UIPicker

UIDatePicker

UISegmentedController

UISwitch

Comunicación

Delegate

Completion Handlers

NotificationCenter

KVO

Observer