

Notificaciones Locales

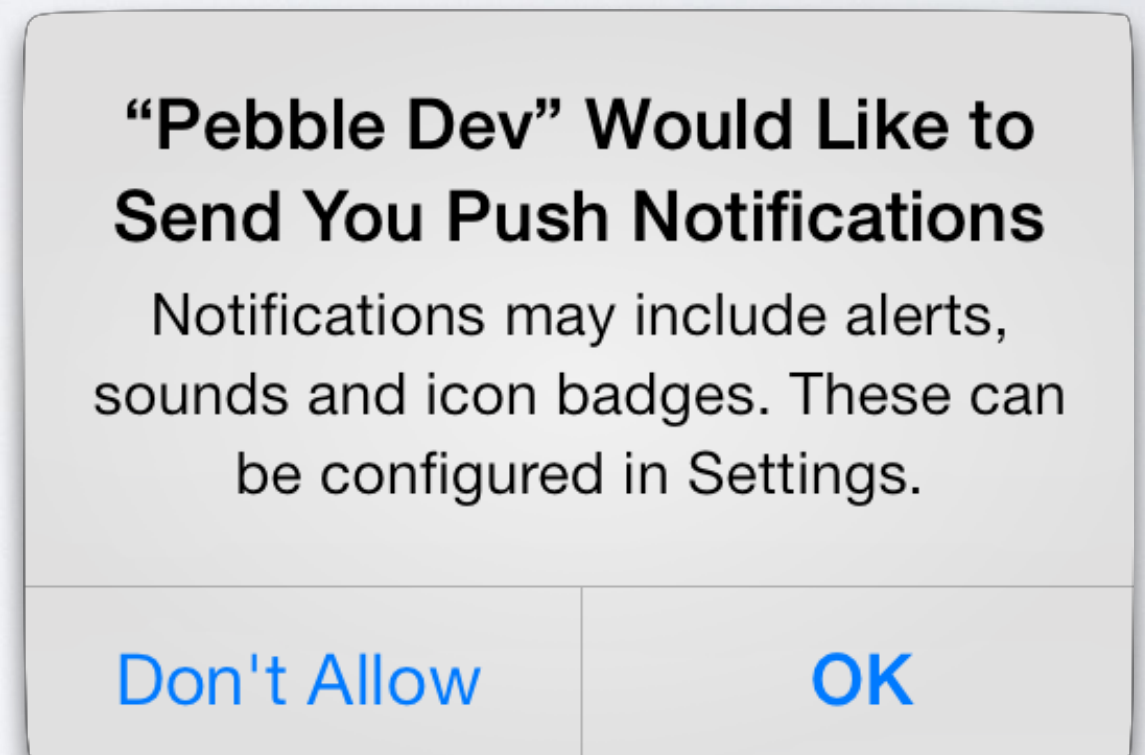
Introducción

Permiten a una aplicación enviar información al usuario en algún momento del futuro, incluso si la aplicación se encuentra cerrada en el momento de la entrega de la notificación.

La notificación aparecerá como una alerta o bien en el centro de notificaciones del dispositivo. Según como hayamos configurado nuestra notificación, y dependiendo de la intervención del usuario, nuestra aplicación podrá ser avisada de la entrega de la notificación.

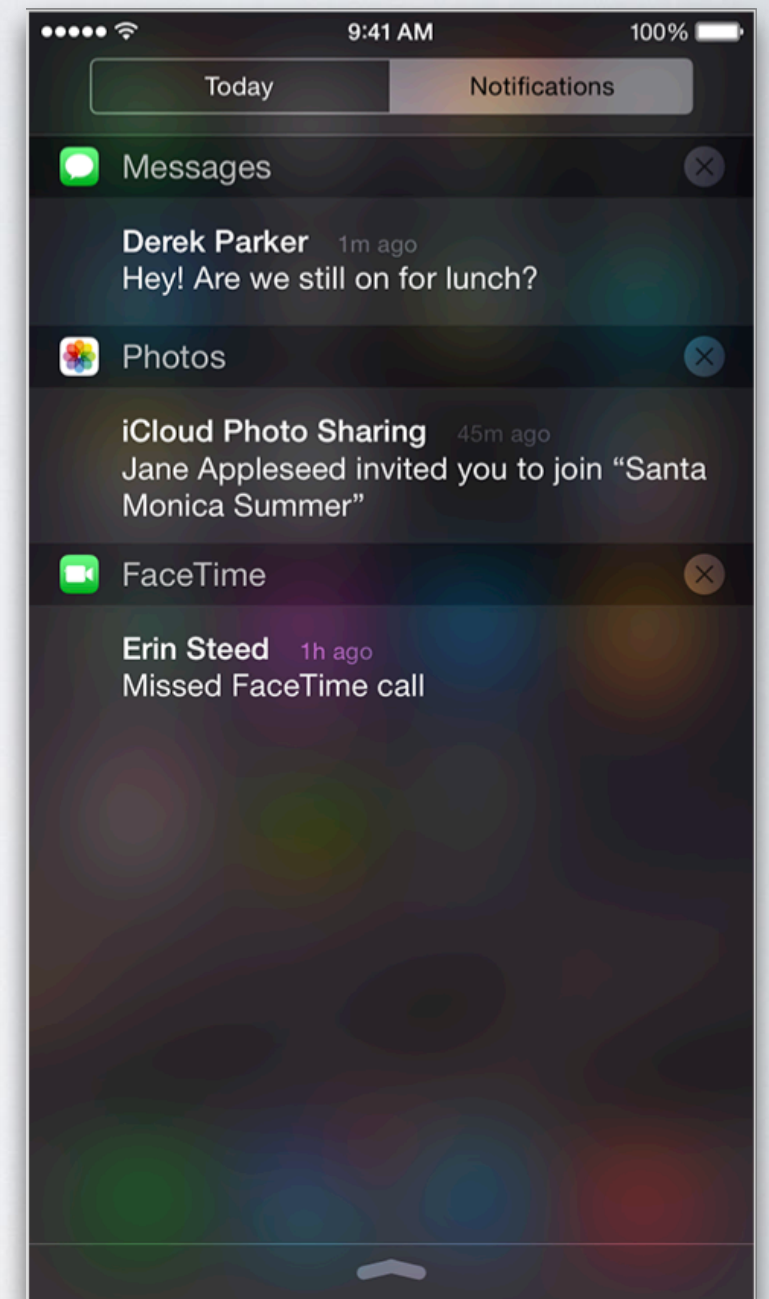
Requerimientos

- Las notificaciones locales son offline.
- Informan de algo programado en la app.
- Se perciben como las remotas (push).



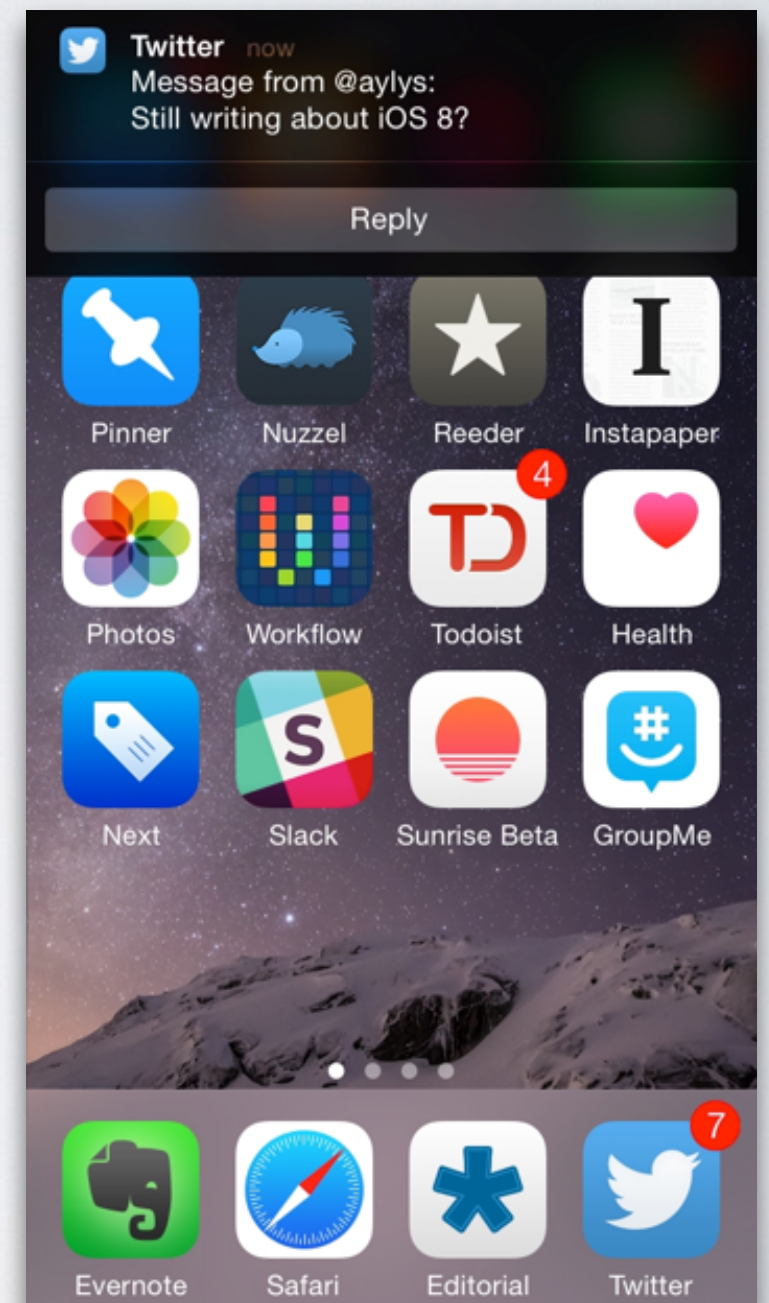
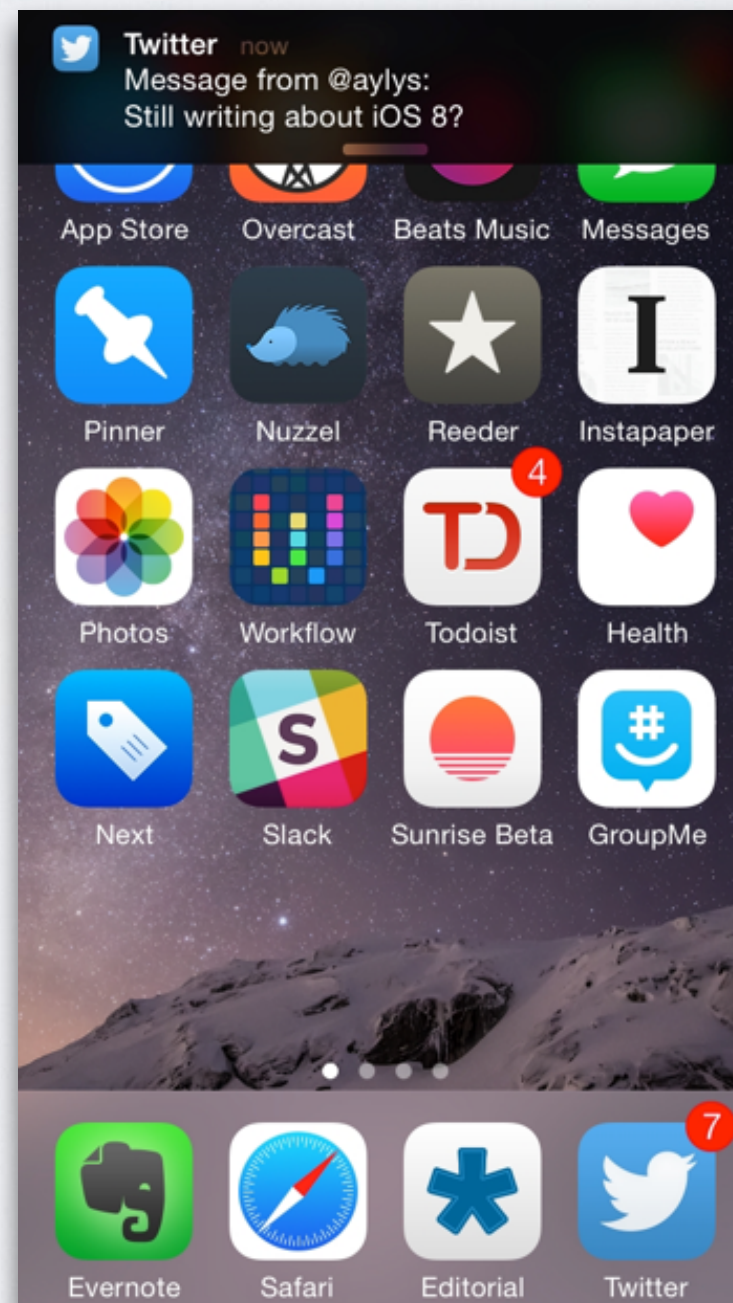
¿Dónde podemos ver las notificaciones?

Centro de notificaciones



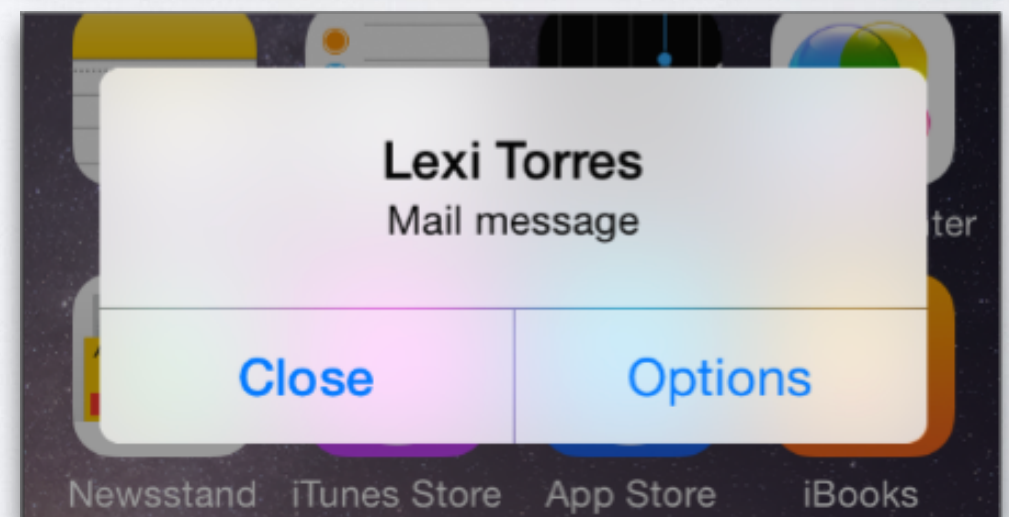
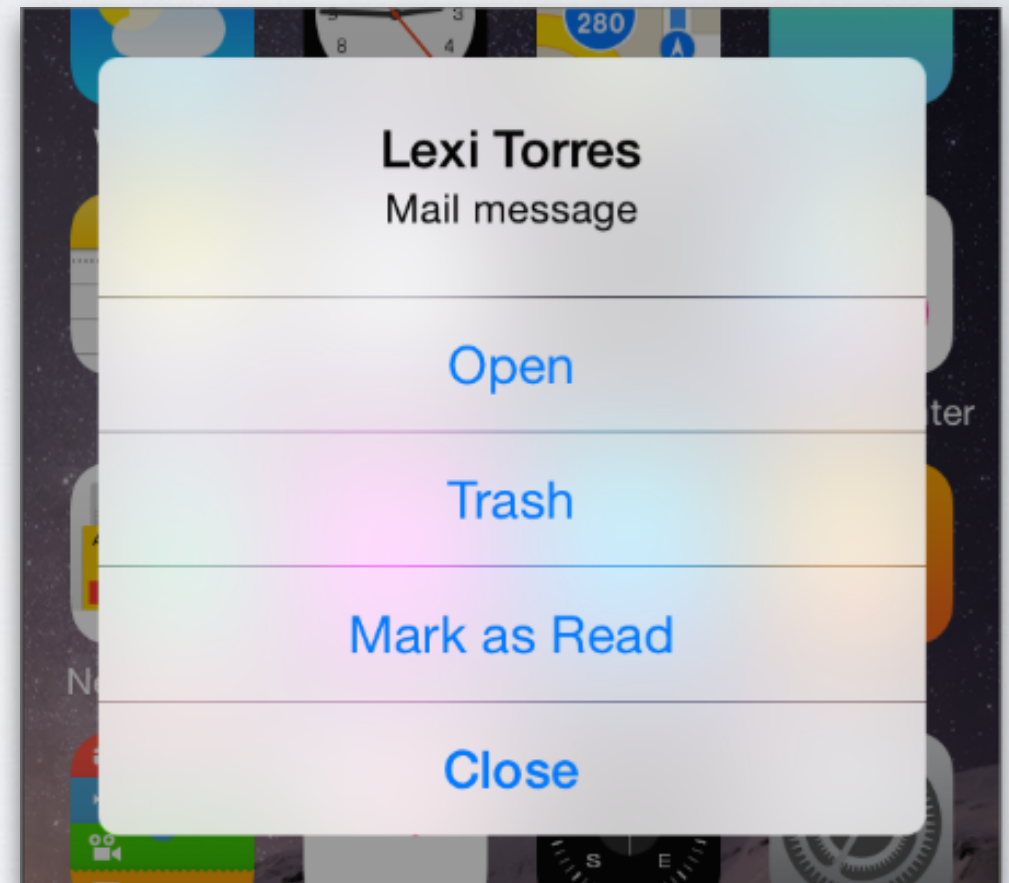
¿Dónde podemos ver las notificaciones?

Alert Banner



¿Dónde podemos ver las notificaciones?

Alert View



UserNotifications Framework

Desde iOS 10 hay un nuevo Framework para notificaciones (locales y push).

Ahora se pueden crear extensiones para mejorar la experiencia de usuario de las notificaciones.

Para apps compatibles con iOS 9 o anteriores, utilizar **UILocalNotification**.

UNNotificationRequest

Un objeto **UNNotificationRequest** especifica una notificación que una aplicación puede programar para su presentación en una fecha y hora específica.

El sistema operativo es el responsable de la entrega de la notificación a la hora programada (no se requiere de servidores remotos).

UNNotificationRequest

- *identifier*: Identificador único. Permite cancelar la notificación más adelante.
- *content*: Instancia de UNNotificationContent con los datos.
- *trigger*: Condición que provoca que se lance la notificación.

UNNotificationContent

- *body*: Mensaje que muestra la notificación.
- *sound*: Sonido de la notificación.
- *title*: Pequeña descripción del por qué de esa alerta.
- *categoryIdentifier*: Nombre de un grupo de acciones a mostrar en la alerta.

UNNotificationContent

- *threadIdentifier*: Para agrupar notificaciones de una misma conversación, por ejemplo.
- *badge*: Badge que se muestra en el icono de la app, indicando el número de notificaciones no vistas por el usuario.
- *userInfo*: Diccionario que contiene información adicional que se le quiera pasar a la app.

UNCalendarNotificationTrigger

- Permite lanzar notificaciones en fechas con unas determinadas características.
- *repeats*: Indica si la notificación es recurrente o no.
- *dateComponents*: Componentes de una determinada fecha.

Ejemplo: usar día y mes de la fecha actual, con *repeats=true*. La notificación se lanzará el 21 de Enero cada año.

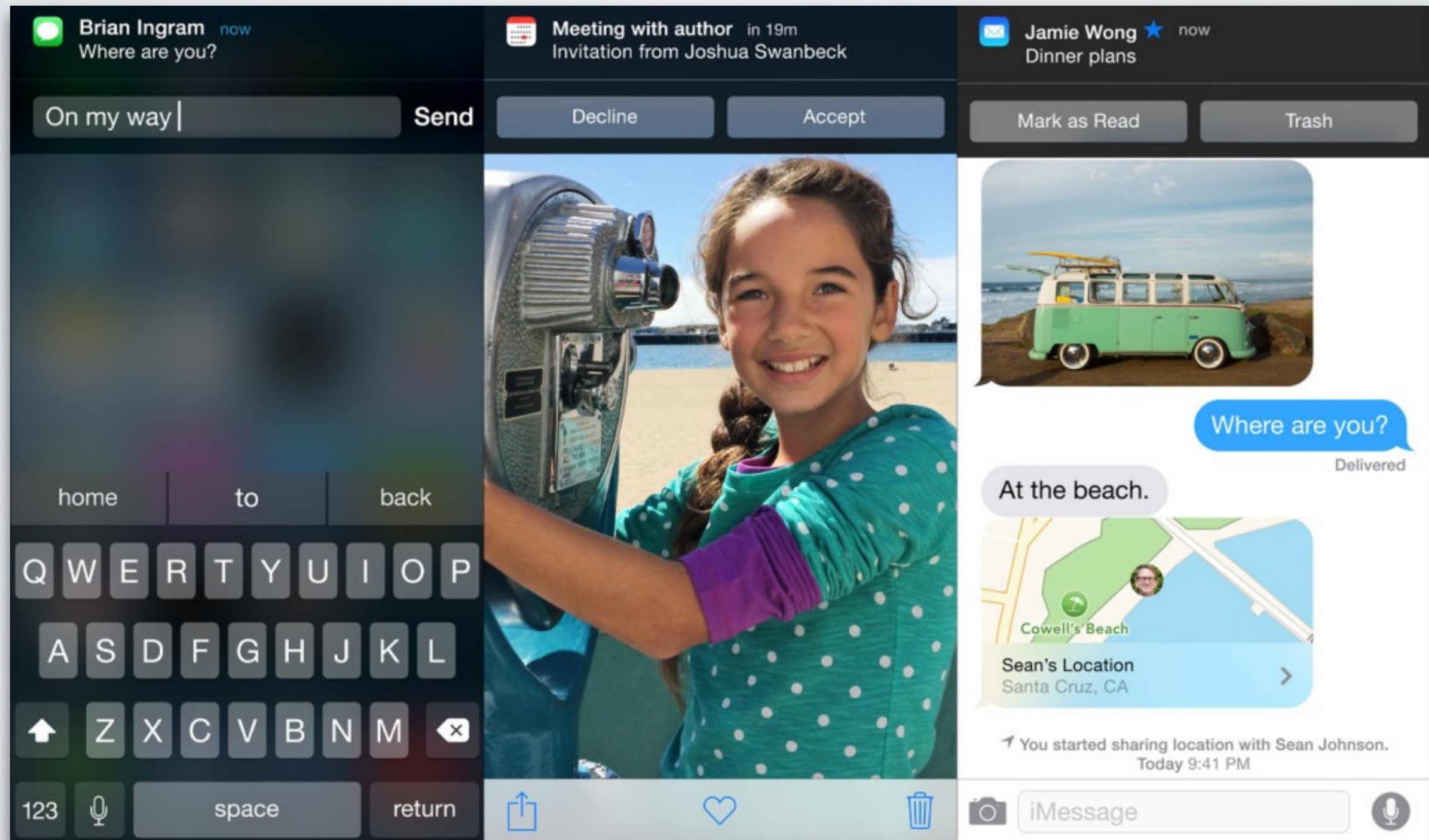
UNNotificationAction

Cuando se entrega una notificación, el sistema muestra un botón para cada acción personalizada asociada a la notificación. Al tocar un botón se lanza la app (ya sea en primer o segundo plano) y lleva a cabo la acción indicada.

UNNotificationAction

- *identifier*: Identificador de la acción.
- *title*: Título del botón que dispara la acción.
- *options*:
 - *.authenticationRequired*: El usuario debe desbloquear el móvil antes de realizar la acción.
 - *.destructive*: Es una acción destructiva.
 - *.foreground*: Esta acción pone la app de vuelta en primer plano.

UNNotificationCategory

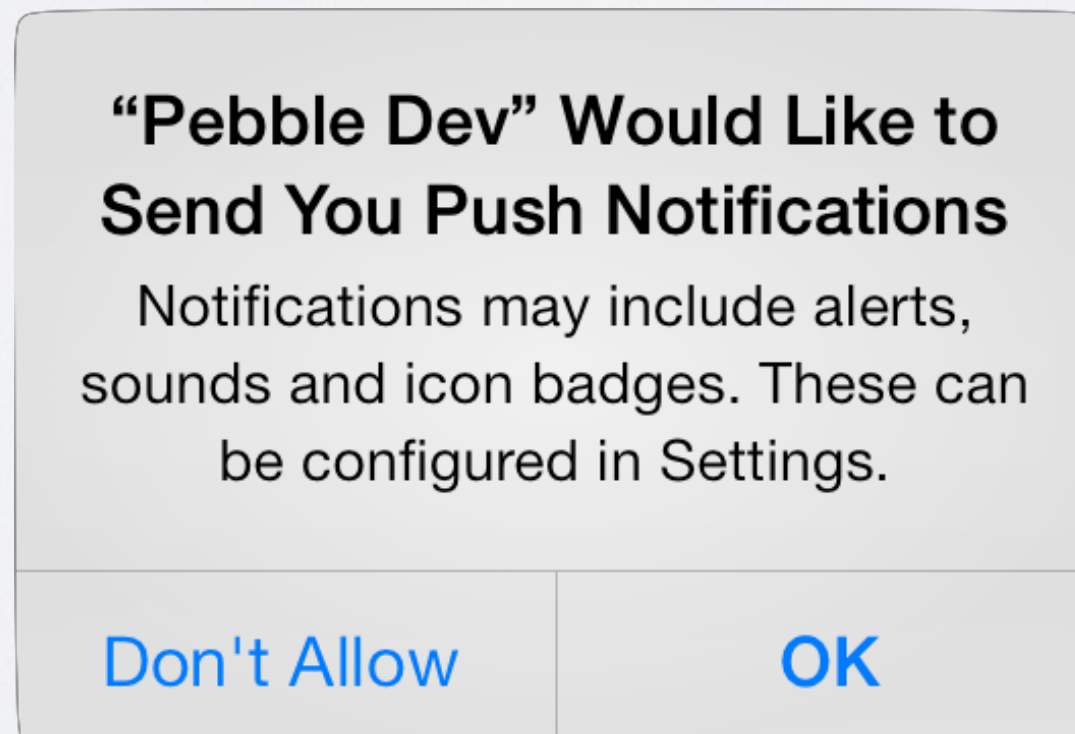


Proceso

1. Registrar notificaciones.
2. Programar notificaciones.
3. Manejar notificaciones recibidas.

Registro de notificaciones

La app **DEBE** pedir permiso al usuario para mostrar notificaciones, indicándole los tipos de notificaciones que habrá.



Registro de notificaciones

```
let notificationCenter = UNUserNotificationCenter.current()
```

```
// Register notification categories
```

```
notificationCenter.setNotificationCategories([...])
```

```
// Request notification permissions
```

```
notificationCenter.requestAuthorization(options: [.alert, .sound, .badge]) {
```

```
    (granted, error) in
```

```
    //Parse errors and track state
```

```
}
```


Programar notificaciones

1. Configurar **UNMutableNotificationContent**.
 1. Asignar *body, categoryId, sound...*
2. Configurar **UNCalendarNotificationTrigger**.
3. Configurar **UNNotificationRequest**.
 1. Generar un identificador único.
4. Programar en **UNUserNotificationCenter** con **addRequest:withCompletionHandler:**.

Manejar notificaciones recibidas

- Protocolo **UNUserNotificationCenterDelegate**.
userNotificationCenter:
didReceiveNotificationResponse:
- Si la app está en segundo plano: se muestra la notificación.
 - Usuario toca el botón por defecto e inicia la app.
 - Se llama al delegado.
- Si la app está en primer plano: sólo se llama al delegado.

Manejo de notificaciones: En primer plano

```
func userNotificationCenter(_ center: UNUserNotificationCenter,  
    willPresent notification: UNNotification,  
    withCompletionHandler completionHandler:  
        @escaping (UNNotificationPresentationOptions) -> Void) {  
  
    // Called when a notification is delivered to a foreground app.  
    completionHandler(.alert)  
  
}
```

Manejo de notificaciones:

Acciones

```
func userNotificationCenter(_ center: UNUserNotificationCenter,  
    didReceive response: UNNotificationResponse,  
    withCompletionHandler completionHandler: @escaping () -> Void) {  
  
    // Called to let your app know which action was  
    // selected by the user for a given notification.  
  
    // App will terminate if we fail to call this  
    completionHandler()  
}
```