

Guía Práctica 3

Eliminar tareas

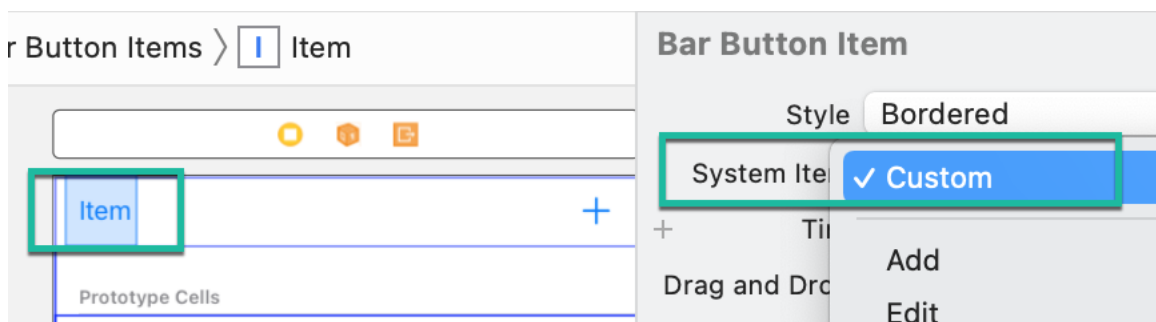
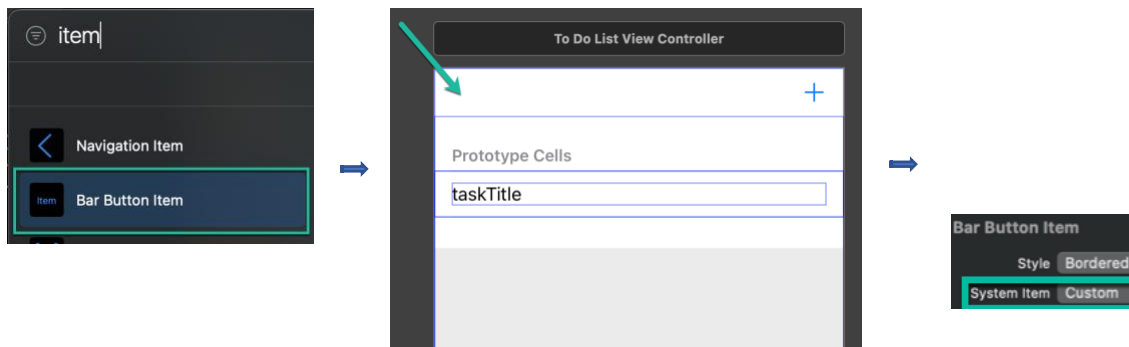
Ya tenemos implementado:

Create: Agregar tareas

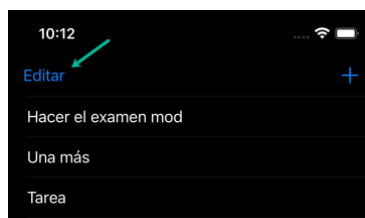
Read: Leer tareas (mostrar el listado de tareas)

Update: Actualizar un tarea en particular

Falta agregar la funcionalidad de eliminar, la D (delete) de nuestro CRUD. En este caso vamos a implementar la eliminación de las actividades de nuestra lista de tareas. Para ello vamos a agregar un botón de tipo "Bar button Item" "Custom", con el título de "Editar" en la vista de nuestro "To Do List Controller"



El nuevo botón debe verse así:

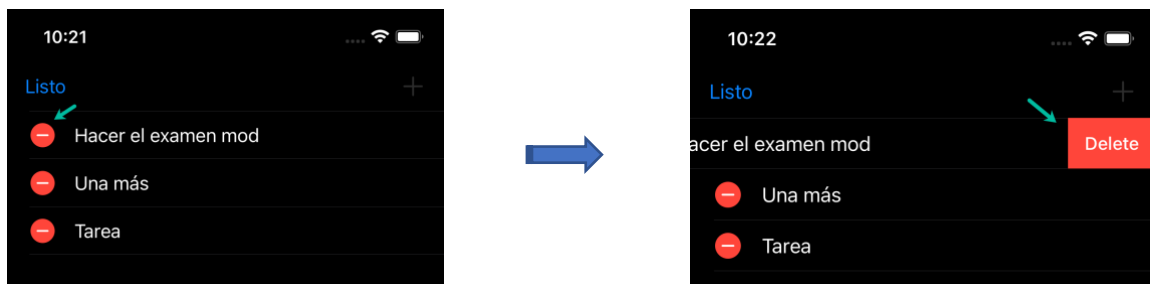


Después agregar el botón, es necesario agregar su IBOutlet (referencia del elemento gráfico en el código) al controller "ToDoListViewController", y una nueva IBAction que llamaremos "editTaskButton" (método que se ejecutará al momento de presionar el botón), al agregar al siguiente código en el IBAction:

```
@IBAction func editTaskButton(_ sender: UIBarButtonItem) {  
    if toDoListTableView.isEditing{  
        toDoListTableView.setEditing(false, animated: true)  
        sender.title = "Editar"  
        addTaskButton.isEnabled = true  
    }  
    else {  
        // ...  
    }  
}
```

Nota: Es posible que xcode no permita crear la IBAction en la sección de la extensión, si fuera el caso la creamos en el controlador y después la cortamos y pegamos a la sección de la extensión del controlador.

Con este cambio ya se muestra el modo de edición para eliminar las tareas de la lista, sin embargo, aun no funcionan



Para que funcione la opción de eliminar que agregar el siguiente código utilizando el método **tableView(:commit:forRowAt)** (Este método permite confirmar la eliminación de un renglón en nuestro table view) en nuestro "ToDoListViewController", recordemos que en él implementamos los métodos asociados a nuestra table view que en la que se muestran las actividades

```
func tableView(_ tableView: UITableView, commit editingStyle:
UITableViewCellEditingStyle, forRowAt indexPath: IndexPath) {
    //si el renglon seleccionado esta en modo de eliminacion
    if editingStyle == .delete{
        //guardamo la tarea seleccionada en la variable "currentTask"
        currentTask = self.tasks[indexPath.row]
        //eliminamos la tarea
        self.context.delete(currentTask!)
        //actualizamos el contexto utilizando el método save
        do{
            try self.context.save()
        }
        catch{
            print("Error: ", error)
        }
        let dataManager = TaskDataManager(context: context)
        //si todo fue bien recuperamos el listado de tareas actualizado
        tasks = dataManager.fetchTasks()
    }
}
```