



Introdução ao Objective-C

E ao SDK do iOS 8

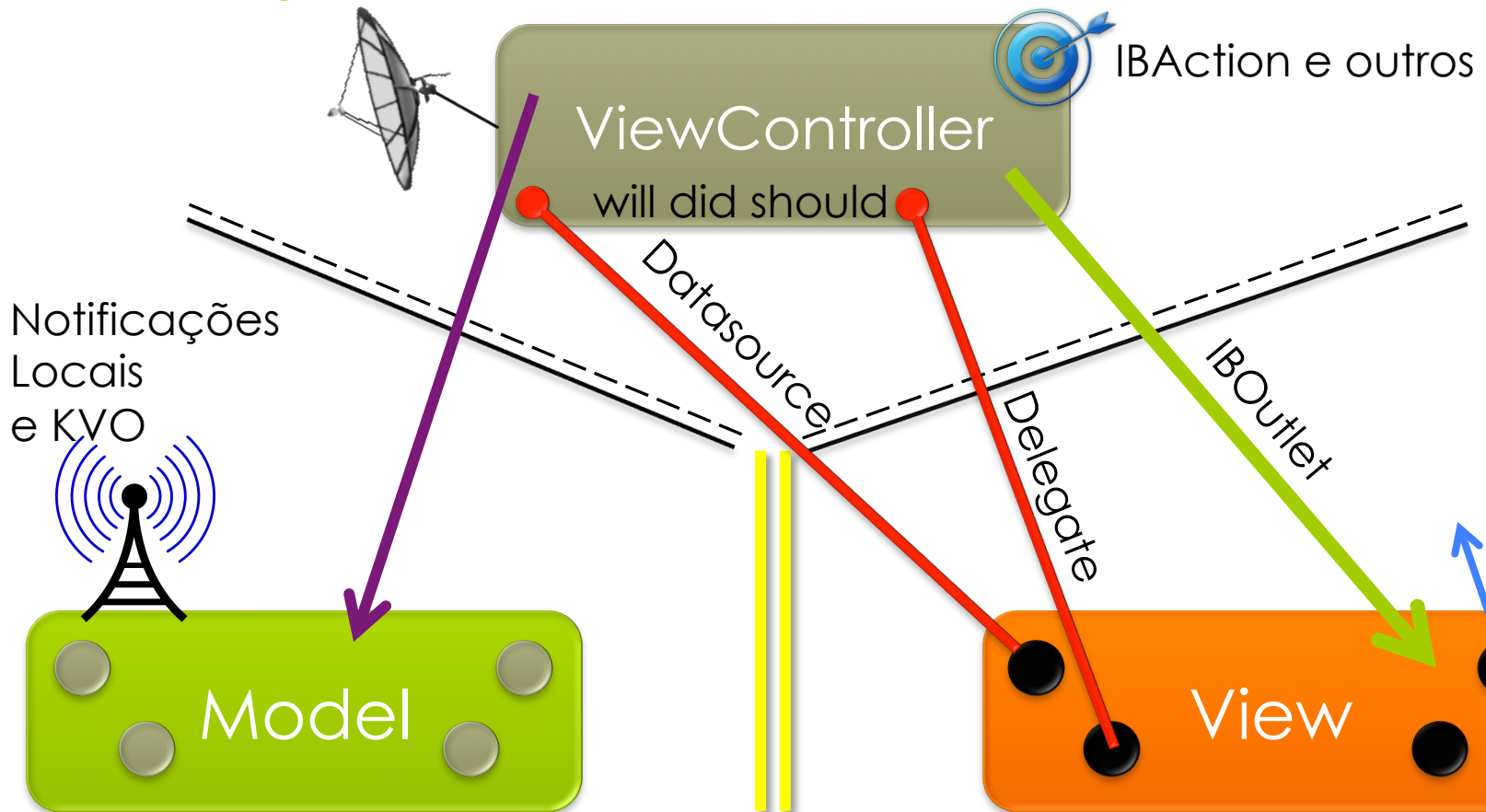
Agenda

- Arquitetura MVVC;
- Outlet;
- Delegate;
- Data source;
- Protocol;
- Hora de Brincar!
 - Mudança de tela;
 - UITableViewController.

Objetivos do dia

- Entender o funcionamento da arquitetura MVVC do iOS
 - Entender a comunicação entre as camadas através de protocolos (Data source, Delegate) e Outlets;

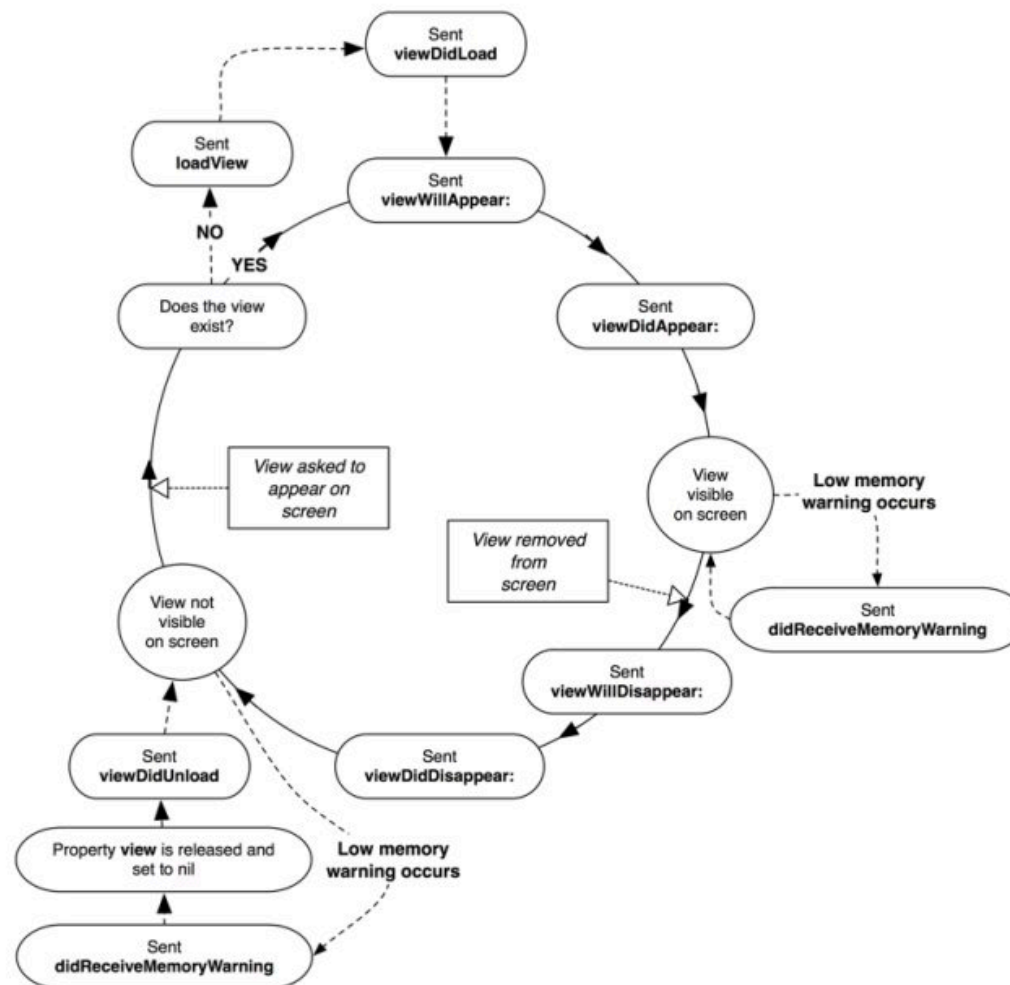
Arquitetura MVVC



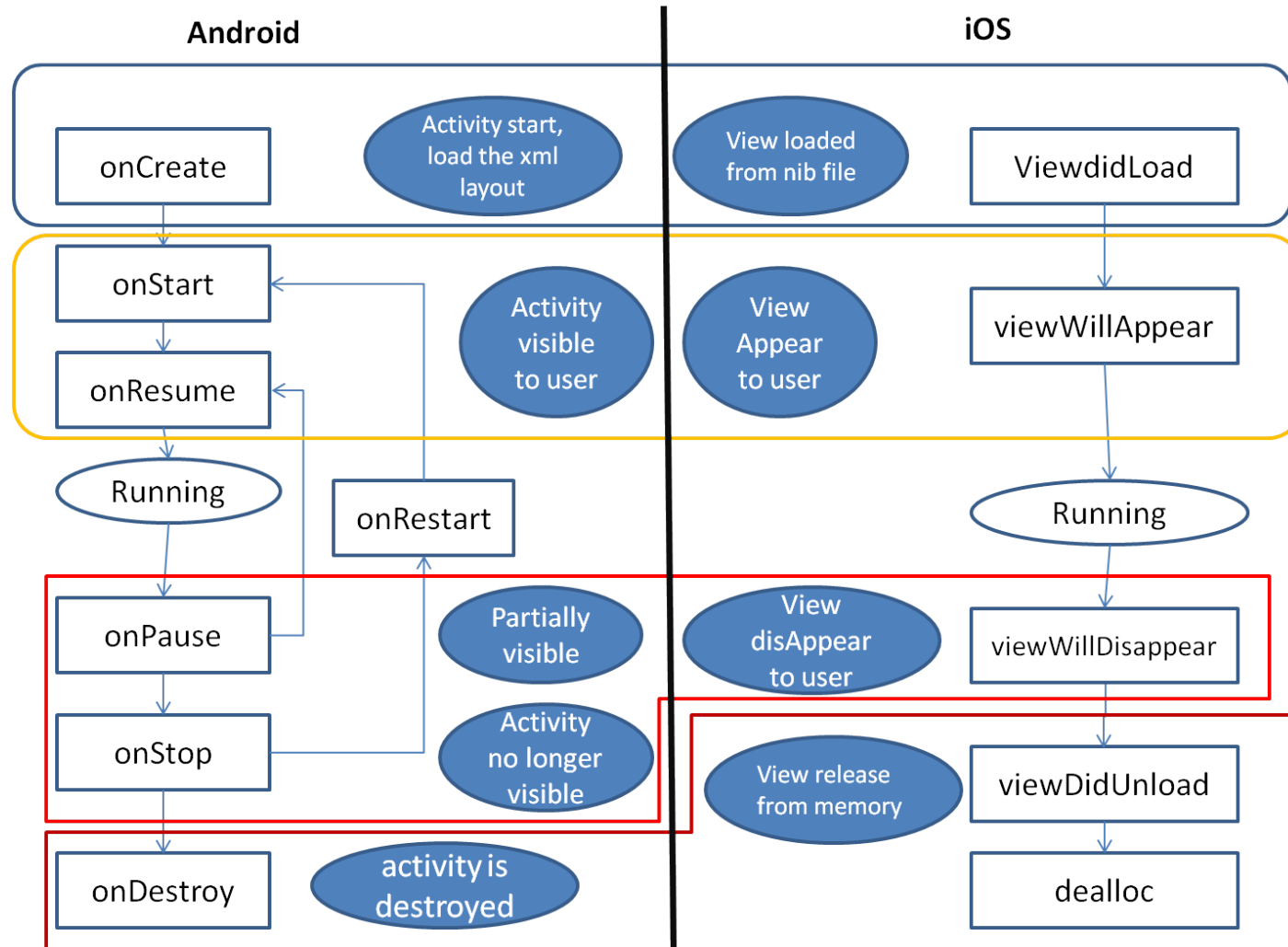
Links Legais!

- Sobre KVO (Key-Value Observing)
 - <http://nshipster.com/key-value-observing/>
- Padrão Delegate
 - http://en.wikipedia.org/wiki/Delegation_pattern
- View Controller
 - https://developer.apple.com/library/ios/featuredarticles/ViewControllerPGforiPhoneOS/AboutViewControllers/AboutViewControllers.html#//apple_ref/doc/uid/TP40007457-CH112-SW10

Ciclo de Vida do ViewController



Ciclo de Vida do ViewController



Protocol

- Define um protocolo de comunicação entre as camadas, de modo a minimizar o acoplamento entre as mesmas;
- Promove, portanto, uma comunicação cega. Lembrar do conceito de troca de mensagens.


```

271 @protocol UITableViewDataSource<NSObject>
272
273 @required
274
275 - (NSInteger)tableView:(UITableView *)tableView numberOfRowsInSection:(NSInteger)section;
276
277 // Row display. Implementers should *always* try to reuse cells by setting each cell's reuseIdentifier and querying for available reusable cells with
278 // Cell gets various attributes set automatically based on table (separators) and data source (accessory views, editing controls)
279
280 - (UITableViewCell *)tableView:(UITableView *)tableView cellForRowAtIndexPath:(NSIndexPath *)indexPath;
281
282 @optional
283
284 - (NSInteger)numberOfSectionsInTableView:(UITableView *)tableView; // Default is 1 if not implemented
285
286 - (NSString *)tableView:(UITableView *)tableView titleForHeaderInSection:(NSInteger)section; // fixed font style. use custom view (UILabel) if you want something else
287 - (NSString *)tableView:(UITableView *)tableView titleForFooterInSection:(NSInteger)section;
288
289 // Editing
290
291 // Individual rows can opt out of having the -editing property set for them. If not implemented, all rows are assumed to be editable.
292 - (BOOL)tableView:(UITableView *)tableView canEditRowAtIndexPath:(NSIndexPath *)indexPath;
293
294 // Moving/reordering
295
296 // Allows the reorder accessory view to optionally be shown for a particular row. By default, the reorder control will be shown only if the datasource implements -canMoveRowAtIndexPath.
297 - (BOOL)tableView:(UITableView *)tableView canMoveRowAtIndexPath:(NSIndexPath *)indexPath;
298
299 // Index
300
301 - (NSArray *)sectionIndexTitlesForTableView:(UITableView *)tableView; // return list of section titles
302 - (NSInteger)tableView:(UITableView *)tableView sectionForSectionIndexTitle:(NSString *)title atIndex:(NSInteger)index; // tell table which section to scroll to
303
304 // Data manipulation - insert and delete support
305
306 // After a row has the minus or plus button invoked (based on the UITableViewCellEditingStyle for the cell), the dataSource must commit the change
307 // Not called for edit actions using UITableViewRowAction - the action's handler will be invoked instead
308 - (void)tableView:(UITableView *)tableView commitEditingStyle:(UITableViewCellEditingStyle)editingStyle forRowAtIndexPath:(NSIndexPath *)indexPath;
309
310 // Data manipulation - reorder / moving support
311
312 - (void)tableView:(UITableView *)tableView moveRowAtIndexPath:(NSIndexPath *)sourceIndexPath toIndexPath:(NSIndexPath *)destinationIndexPath;
313
314 @end

```