

Change Cover

	Extension	ViewLogic	DetailViewController	Model Logic	ModelPersistence	ScoreTableViewCo ntroller
Acceso a la biblioteca de fotos			<i>IBAction changeCover</i> configuration PHPickerViewController present delegate			
Patrón delegate para seleccionar y cargar la imagen		<i>func getImageFromLibrary(results: [PHPickerResult,callback...]</i> guard let providers. = results.map... RunLoop.main.perform Callback ->	Conformarlo con PHPickerViewControllerDelegate <i>func picker</i> dismiss viewLogic.getImageFromLibrary -> callback			
Actualizar score a json local				<i>var scores:[score]</i> didSet try? persistence.saveScores(score:s cores)	<i>func getScores</i> if FileManager.default.fileExists Local else Bundle	
Escalar tamaño imagen	UIImage <i>func resizeImage -> UIImage?</i> image.preparing/prepare/byPreparing Thumbnail	<i>func getImageFromLibrary</i> ... image.resizeImage ...				
Grabar imagen seleccionada a disco					<i>func saveImage(id, image urlImage... let urlImage... If let data.compressionQuality... try data.write...</i>	
			<i>var newCover:UIImage?</i> didSet If let newCover cover.image = newCover ... <i>func picker</i> ... <i>self.newCover.image = image</i>	<i>func updateScore(score:Score, newCover:UIImage?)</i> ... If let newCover try? persistence.saveImage		
Carga de la imagen de disco y mostrarla en la tableView					<i>func loadImage(id:Int) throws -> UIImage?</i> let urlImage = URL.document... If fileExists... return UIImage(data... else nil	
		<i>let persistence = ModelPersistence.shared</i> ... <i>func getCover(score:Score)->UIImage?</i> If let cover = try? persistence.loadImage(id:score.id) return cover else return UIImage(named:score.cover) <i>func cellForRowAt...</i> ... content.image = getCover(score... content.imageProperties.maximumSize = CGSize(width:100...	<i>super.viewDidLoad()</i> ... If let selected = selectedScore ... cover.image = viewLogic.getCover(selected...			<i>func tableView... cellForRowAt...</i> ... content.image = viewLogic.getCover (score...