

FACULDADE DE ENGENHARIA DA UNIVERSIDADE DO PORTO



FEUP FACULDADE DE ENGENHARIA
UNIVERSIDADE DO PORTO

iPortalDoc Mobile

Nuno Filipe Pinto Ferreira

DISSERTAÇÃO

MESTRADO INTEGRADO EM ENGENHARIA ELECTROTÉCNICA E DE
COMPUTADORES

Orientador Interno: Luis Miguel Pinho de Almeida

Orientador Externo: Telma Cristina Marques Salgueiro

25 de Junho de 2017

Resumo

O iPortalDoc é um Sistema de Gestão Documental e Workflow (SGDW) que funciona como um serviço de valor acrescentado para a Intranet e está integrado no Sistema Operacional IPBrick. O iPortalDoc utiliza como Sistema de Gestão de Base de Dados (SGBD) o PostgreSQL. O iPortalDoc permite o registo, classificação e tratamento de todas as comunicações (e-mails, conversações de Instant Messaging, chamadas telefónicas, faxes), bem como o tratamento de todos os processos de uma organização. Tendo em conta o aumento do número de utilizadores que recorrem aos dispositivos móveis para acederem ao iPortalDoc e consultarem documentos ou executarem as suas ações pendentes, pretende-se disponibilizar a este tipo de utilizadores a versão móvel, chamada iPortalDoc Mobile. Este trabalho aborda o problema de definir e desenvolver uma interface móvel enquanto são cumpridos os requisitos da aplicação, seguindo os princípios e soluções encontradas de User Experience e User Interface. A dissertação inclui uma estudo geral desses princípios, uma discussão sobre a sua aplicação no desenvolvimento do iPortalDoc e a definição dos seus requisitos. A dissertação engloba os processo de desenvolvimento, descrevendo a estrutura e funcionamento da aplicação. A última parte inclui um resumo dos resultados práticos que validam a aplicação face aos seus requisitos.

Abstract

iPortalDoc is a Document Management and Workflow System, which works as a value-added service for the Intranet and it is integrated in the IPBrick Operating System. IPortalDoc uses the PostgreSQL Database Management System (DBMS). IPortalDoc allows the registration, classification and treatment of all communications (e-mails, instant messaging, telephone calls, faxes) as well as the procedural treatment of all the processes of an organization. Given the increased number of users who are turning to mobile devices to access iPortalDoc and perform some of their tasks or check documents, there is a need to make a mobile version available to this type of users, named iPortalDoc Mobile. This work addresses the problem of defining and developing a mobile interface while meeting the requirements for the application, following the principles and solutions of User Experience and User Interface. This dissertation includes a general survey of these principles, a discussion on their application to the development of iPortalDoc Mobile and the definition of its requirements. The dissertation then covers the development process, describing the structure and operation of the application. The last part includes a summary of practical results that validate the application against its requirements.

Agradecimentos

Esta secção é dedicada e serve para agradecer a todas as pessoas que deram a sua contribuição, grande ou pequena, e tornaram a realização desta dissertação possível.

Gostaria de agradecer em primeiro lugar à IPBRICK, SA, pelo projeto desafiante e fascinante e pela oportunidade de o desenvolver nas suas instalações com todas as condições necessárias para realizar um bom trabalho.

À Eng. Telma Salgueiro da IPBRICK, SA pela aposta em alguém que embora com muita vontade de aprender, não tinha experiência e créditos firmados no âmbito desta dissertação.

Ao Prof. Luis Almeida pela disponibilidade que mostrou durante toda o período da dissertação, no esforço desenvolvido na leitura, nas sugestões e orientação no contexto mais formal da dissertação.

Ao Eng. Rui Carvalho da IPBRICK, SA, o responsável pelo projeto que esteve na base desta dissertação, por todo o trabalho, paciência, ajuda e transmissão de conhecimento constantes desde o início ao fim do período de dissertação.

À equipa da IPBRICK, SA por permitir uma fácil integração e adaptação à empresa.

Aos meus amigos e família, principalmente os meus pais, por todo o apoio e motivação ao longo do curso.

À minha namorada, um agradecimento especial, pelo ânimo, compreensão e por me ajudar a relativizar os problemas.

Por último, agradeço a todos aqueles que não foram mencionados, mas que de uma forma ou outra também contribuíram para a elaboração deste trabalho.

Nuno Ferreira

*“You, me, or nobody is gonna hit as hard as life,
but it ain’t about how hard you hit,
it’s about how hard you can get hit and keep moving forward,
how much you can take and keep moving forward.
That’s how winning is done.”*

Rocky Balboa

Conteúdo

1	Introdução	1
1.1	Motivação	2
1.2	Objetivos	2
1.3	Estrutura do Relatório	2
2	Caracterização do Problema	5
2.1	Definição do Problema	5
2.2	Abordagem ao Problema	6
3	Revisão Bibliográfica	11
3.1	User eXperience (UX)	11
3.1.1	Definição	11
3.1.2	Decomposição	11
3.1.3	Gamification	12
3.2	User Interface (UI)	13
3.2.1	Definição	13
3.2.2	Princípios	13
3.2.3	Alternativas	16
3.3	Frameworks	16
3.3.1	Frameworks de desenvolvimento de aplicações móveis	16
3.3.2	Frameworks de User Interface de aplicações móveis	18
3.3.3	Frameworks PHP	20
3.4	Conclusão	22
4	A plataforma iPortalDoc	23
4.1	iPortalDoc	23
4.1.1	Interface	24
4.1.2	Casos de Uso	26
4.2	Modelo e Requisitos do iPortalDoc Mobile	27
4.2.1	Login	27
4.2.2	Menu	28
4.2.3	Lista de ações	28
4.2.4	Ação	29
4.2.5	Informação da Ação	30
4.2.6	Esquema Cronológico da Ação	31
4.3	Conclusão	32

5	A solução iPortalDoc Mobile	33
5.1	Estrutura	33
5.1.1	Frontend	33
5.1.2	Backend	35
5.2	Escolha das Frameworks	35
5.2.1	Apache Cordova	36
5.2.2	Ionic	37
5.2.3	Yii	38
5.3	Arquitetura	38
5.3.1	Alto Nível	38
5.3.2	Baixo Nível	42
5.4	Implementação	42
6	Resultados do iPortalDoc Mobile	49
6.1	Refinamento da solução	49
6.2	Resultados de Implementação	50
6.3	Resultados de User Experience	58
6.4	Análise do cumprimento requisitos	61
6.5	Experiências e análise crítica global ao iPortalDoc Mobile	62
7	Conclusão e trabalhos futuros	63
7.1	Trabalho Futuro	63
A	Diagrama de Classes	65
B	Diagramas de Sequência	69
	Referências	91

Lista de Figuras

1.1	Aplicações móveis [1]	1
2.1	Adaptação de uma aplicação Web para mobile [2]	5
2.2	Abordagem empírica para a lista de casos de uso da aplicação móvel	6
2.3	Possibilidade para a página de login do iPortalDoc Mobile	7
2.4	Possibilidades para a página frontal do iPortalDoc Mobile	8
2.5	Possibilidade para a página frontal de pesquisa	8
4.1	Hierarquia de Documentos	24
4.2	Documentos	24
4.3	Barra de Ferramentas	25
4.4	Pesquisa	25
4.5	iPortalDoc - Casos de Uso	26
4.6	Modelos para a página Login	27
4.7	Modelos para a página Lista de ações	29
4.8	Modelo para a página Ação	30
4.9	Modelo para a página Info	31
4.10	Modelo para a página de Esquema cronológico	31
5.1	Arquitetura do Apache Cordova [3]	36
5.2	Exemplo de utilização da Ionic [4]	37
5.3	Diagrama de Classes - Modelo de especificação	40
5.4	Diagrama de Sequência - Início da aplicação	41
5.5	Exemplo da implementação de um ficheiro de template HTML	44
5.6	Exemplo da implementação de um ficheiro de estilo SCSS	45
5.7	Exemplo da implementação de um ficheiro de controlo TS	46
5.8	Exemplo da implementação de um serviço em PHP	47
5.9	Exemplo da implementação de um controlador da API em PHP	48
6.1	Modelo da página de documentos	49
6.2	Resultados da página de Documentos	50
6.3	Comparação entre o resultado e o modelo da página de Login	51
6.4	Comparação entre o resultado e o modelo da página de Login com informação	51
6.5	Resultado da implementação da página Home	52
6.6	Resultado da implementação da página Menu	52
6.7	Comparação entre o resultado e o modelo da página de Lista de Ações	53
6.8	Comparação entre o resultado e o modelo da página de Ação	53
6.9	Comparação entre o resultado e o modelo da página de Info	54
6.10	Comparação entre o resultado e o modelo da página de Esquema Cronológico	54

6.11	Resultado da implementação da página Agendamento	55
6.12	Resultado da implementação da página Marcação	55
6.13	Resultado da implementação da página Lista VIP	56
6.14	Resultado da implementação da página Lista de Comentários Pré-definidos	56
6.15	Resultado da implementação da página Comentário	57
6.16	Resultado da implementação da página Lista de Ausências	57
6.17	Resultado da implementação da página Ausência	58
6.18	Resultado da implementação do conceito de UX: Saídas claras	58
6.19	Resultado da implementação do conceito de UX: Feedback	59
6.20	Resultado da implementação do conceito de UX: Respostas	59
6.21	Resultado da funcionalidade Pesquisas implementadas	59
6.22	Resultado da funcionalidade Localização dos botões de execução implementada	60
6.23	Resultado da implementação para uma resolução de tablet (iPad Air)	60
6.24	Evolução do cumprimento de requisitos funcionais	61
A.1	Diagrama de Classes (1 de 3)	66
A.2	Diagrama de Classes (2 de 3)	67
A.3	Diagrama de Classes (3 de 3)	68
B.1	Diagrama de Sequência - Login (1 de 3)	70
B.2	Diagrama de Sequência - Login (2 de 3)	71
B.3	Diagrama de Sequência - Login (3 de 3)	72
B.4	Diagrama de Sequência - Home	73
B.5	Diagrama de Sequência - Lista de Ações (1 de 5)	74
B.6	Diagrama de Sequência - Lista de Ações (2 de 5)	75
B.7	Diagrama de Sequência - Lista de Ações (3 de 5)	76
B.8	Diagrama de Sequência - Lista de Ações (4 de 5)	77
B.9	Diagrama de Sequência - Lista de Ações (5 de 5)	78
B.10	Diagrama de Sequência - Info (1 de 2)	79
B.11	Diagrama de Sequência - Info (2 de 2)	80
B.12	Diagrama de Sequência - Schedule (1 de 2)	81
B.13	Diagrama de Sequência - Schedule (2 de 2)	82
B.14	Diagrama de Sequência - Chronological Scheme (1 de 2)	83
B.15	Diagrama de Sequência - Chronological Scheme (2 de 2)	84
B.16	Diagrama de Sequência - Mark (1 de 2)	85
B.17	Diagrama de Sequência - Mark (2 de 2)	86
B.18	Diagrama de Sequência - Action (1 de 4)	87
B.19	Diagrama de Sequência - Action (2 de 4)	88
B.20	Diagrama de Sequência - Action (3 de 4)	89
B.21	Diagrama de Sequência - Action (4 de 4)	90

Lista de Tabelas

3.1	Definição das dimensões hierárquicas da UX [5]	12
-----	--	----

Abreviaturas e Símbolos

AJAX	Asynchronous Javascript and XML
API	Application Programming Interface
APP	Application
CSS	Cascading Style Sheets
HTML	HyperText Markup Language
HTTP	HyperText Transfer Protocol
JS	JavaScript
MVC	Model-View-Controller
PHP	PHP Hypertext Preprocessor
SCSS	Sassy Cascading Style Sheets
TS	TypeScript
UI	User Interface
UX	User Experience

Capítulo 1

Introdução



Figura 1.1: Aplicações móveis [1]

Num mundo em constante movimento e contacto, cada vez mais as pessoas recorrem ao dispositivo de comunicação de mais fácil e rápido acesso que têm próximo de si. Atualmente, estes dispositivos são genericamente chamados dispositivos móveis e passaram de meros objetos tecnológicos para objetos sociais chave e ferramentas bastante difundidas [6]. Apareceram mais de 563 milhões novos dispositivos móveis e conexões à Internet em 2015 e os smartphones representam a maior parte desse crescimento, tendo o uso médio desses dispositivos aumentado em 43%. Prevê-se que em 2020, perto de 50% dos dispositivos de comunicações serão móveis e que igualmente 50% da conexões de comunicação serão feitas a partir destes dispositivos [7]. Tarefas como aceder a informações relativamente a uma fatura ou executar tarefas de pouca complexidade serão cada vez mais feitas no telemóvel. Esta tendência leva a que empresas, como a IPBrick que motivou esta dissertação, aumentem a sua aposta em serviços que possam ser oferecidos por esta via, de forma a não estarem em desvantagem relativamente aos seus concorrentes mais diretos trazendo novos produtos para o mercado que possam a satisfazer as necessidades dos seus clientes.

Nesta dissertação vamos analisar o caso do iPortalDoc que é uma aplicação web de um sistema de gestão documental e Workflow da IPBrick e que permite aos seus utilizadores gerir o

fluxo dos documentos de uma empresa [8]. O crescente número de acessos à plataforma via dispositivos móveis para executar algumas tarefas faz com que surja a necessidade de disponibilizar aos seus utilizadores uma versão mais cuidada e pensada tendo em conta as especificidades desses dispositivos como interface de aplicações informáticas.

1.1 Motivação

Esta dissertação foi proposta pela IPBrick e realizada em contexto empresarial com o fim de desenvolver um produto final que possa ser efetivamente utilizado pelos utilizadores do sistema iPortalDoc, possibilitando o uso eficaz e eficiente dessa plataforma de gestão documental a partir de qualquer dispositivo móvel. O iPortalDoc é uma ferramenta de gestão complexa pois disponibiliza uma extensa lista de funcionalidades descritas na subsecção 4.1.2. A partir desta lista foram definidos o modelo para a aplicação móvel a desenvolver, o iPortalDoc Mobile, e que são descritos na secção 4.2. Em termos de funcionalidades, pretende-se implementar apenas um subconjunto das que são disponibilizadas pela aplicação original, utilizando um design adequado, seguindo princípios típicos do desenvolvimento de aplicações móveis tendo em conta que nos dispositivos móveis características como poder de processamento, memória de armazenamento e tamanho do ecrã são significativamente inferiores quando comparadas com as características típicas dos computadores. O resultado esperado é criar uma aplicação que satisfaça o seu propósito de maneira intuitiva e agradável para quem a utiliza.

1.2 Objetivos

Os objetivos específicos desta dissertação são os seguintes:

- Estudar conceitos e princípios de User Experience e User Interface assim como frameworks relacionadas, no contexto das aplicações móveis;
- Definir o modelo e requisitos para a aplicação iPortalDoc Mobile descritos;
- Estruturar a aplicação nas suas partes de frontend e backend;
- Escolher as frameworks a utilizar e projetar a aplicação em alto e baixo nível;
- Implementar o iPortalDoc Mobile tendo sempre em conta os requisitos e os conceitos de User Experience e User Interface;
- Validar o iPortalDoc Mobile em contextos controlados de utilização real.

1.3 Estrutura do Relatório

A parte restante desta dissertação está estruturada da seguinte forma. O capítulo 2 descreve o problema subjacente a esta dissertação e que consiste em desenvolver uma versão móvel para

uma aplicação Web de desktop já existente, o iPortalDoc, adaptando a interface de utilização ao dispositivo móvel que está a ser utilizado. Este capítulo aborda o problema providenciando um exemplo base para o desenvolvimento da solução final.

O capítulo 3 apresenta um estudo dos conceitos relacionados com a experiência do utilizador em aplicações Web de desktop e móveis assim como um levantamento de trabalhos realizados e frameworks utilizadas no contexto do desenvolvimento de aplicações móveis.

O capítulo 4 refere detalhadamente todos os pormenores da aplicação iPortalDoc para desktop, as interfaces e os casos de uso típicos e descreve o modelo e requisitos que vão servir de base para o desenvolvimento do iPortalDoc Mobile.

O capítulo 5 descreve como foi solucionado o problema de construir a aplicação móvel de base, projetando primeiramente a estrutura de frontend e backend, a escolha das frameworks, a arquitetura de alto e baixo nível terminando com a implementação do iPortalDoc Mobile.

O capítulo 6 mostra o refinamento da solução devido ao aparecimento de novos requisitos, os resultados finais de interface obtidos comparativamente aos modelos iniciais e os resultados de User Experience. Este capítulo também apresenta uma avaliação que foi feita à aplicação desenvolvida analisando a evolução do cumprimento de requisitos funcionais do iPortalDoc Mobile.

O capítulo 7 conclui esta dissertação e apresenta algumas linhas para potencial evolução deste projeto.

Capítulo 2

Caracterização do Problema

Este capítulo explica o problema abordado nesta dissertação de adaptar uma aplicação Web de desktop, nomeadamente o iPortalDoc, para uma aplicação móvel cumprindo requisitos mínimos de funcionalidade e design. Como ilustração do problema este capítulo também apresenta uma abordagem empírica ao desenvolvimento da versão móvel do iPortalDoc.

2.1 Definição do Problema



Figura 2.1: Adaptação de uma aplicação Web para mobile [2]

O problema desta dissertação traduz-se na criação da versão móvel da aplicação Web de desktop iPortalDoc. Isto traz implicações para a aplicação visto que um dispositivo móvel difere de um computador em características como poder computacional, memória de armazenamento ou tamanho do ecrã, dessa forma a aplicação nunca poderá ser a mesma. As tarefas a executar não poderão ser de elevada complexidade visto que um dispositivo móvel de telecomunicações não tem poder para executar tarefas que exijam esforço computacional superior à sua capacidade. O espaço de armazenamento tem igualmente que ser reduzido e adaptado aos dispositivos que se pretendem atingir. As aplicações Web na sua generalidade não permitem apenas um tipo de funcionalidade, na realidade são abundantes em interações possíveis com o utilizador, o que obriga a que o número de funções que disponibilizam seja adequado à plataforma em que estão a ser utilizadas.

O problema consiste em estudar conceitos de User Experience, User Interface e frameworks de forma a projetar uma aplicação capaz de cumprir todos os requisitos de funcionamento e de design. O último procedimento resume-se em transformar a plataforma desenvolvida com linguagens de programação Web numa aplicação de forma a ser acedida por qualquer dispositivo móvel.

2.2 Abordagem ao Problema

Identificado o problema da dissertação e antes de se conhecer os requisitos funcionais da aplicação móvel, procedeu-se a uma abordagem ao mesmo. Primeiramente ocorreu um estudo preliminar da aplicação em causa, o iPortalDoc descrito no capítulo 4, para gerar uma seleção empírica das funcionalidades a suportar. Elaborou-se uma lista provisória de casos de uso para a aplicação móvel, representada na figura 2.2.

Search				Legenda			
	Quick search			■	-	Funcionalidade disponível	
	Advanced search			■	-	Indeterminada	
Documents				■	-	Funcionalidade indisponível	
	Navigation area in document hierarchy			■			
	Documents visualization area			■			
	Without Selecting a Document						
	Insert			■			
	Remove			■			
	Link Docs			■			
	Move Docs			■			
	Selecting a Document						
	Actions			■			
	Open			■			
	Insert			■			
	Forward			■			
	Add commento			■			
	Remove			■			
	Update Rev (new edition)			■			
	Edit info			■			
	Associate			■			
	Dissociate			■			
	Link			■			
	Link Documents			■			
	Tag			■			
	Move			■			
	Move Docs			■			
	Permissions			■			
	Print			■			
	Cancel Workflow			■			
	Information			■			
	Doc info			■			
	Assoc Doc			■			
	Assoc Mais			■			
	Editions			■			
	Doc Workflow			■			
	Doc Actions			■			

Figura 2.2: Abordagem empírica para a lista de casos de uso da aplicação móvel

Com base na seleção foi desenvolvido um conjunto de modelos de interfaces que permitiram um primeiro contacto com as tecnologias envolvidas e problemas subjacentes. Em relação ao design da aplicação existem várias abordagens possíveis sobre o tamanho, cor, tipo de letra ou ícones por exemplo, no entanto seguiu-se o princípio que refere a importância de minimizar as diferenças comparativamente ao aspeto da versão desktop, para que não haja um equívoco e se transmita uma sensação errada aos utilizadores. As cores da aplicação têm um papel primordial na transmissão de sensações de conforto e orientação [9].

Em primeiro lugar foi elaborado um modelo da página de login a ser mostrada aos utilizadores quando acedem ao iPortalDoc, que está representada na figura 2.3. Uma página minimalista com o nome da aplicação e os campos para login, com a mesma cor da versão desktop para que não haja discrepâncias e não se correr o risco da aplicação parecer outra.

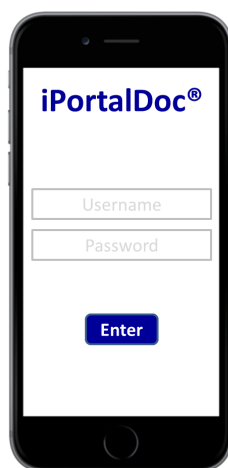


Figura 2.3: Possibilidade para a página de login do iPortalDoc Mobile

Depois de elaborada a página de login foi projetada a página frontal da aplicação que recebe os utilizadores depois destes inserirem credenciais válidas. Para elaborar o layout desta página têm que ser consideradas diversas questões e terão que ser feitas escolhas quer a nível de conteúdo quer de forma.

Foram desenvolvidos dois mockups, simbolizando duas opções para a página frontal da aplicação, representados na figura 2.4.

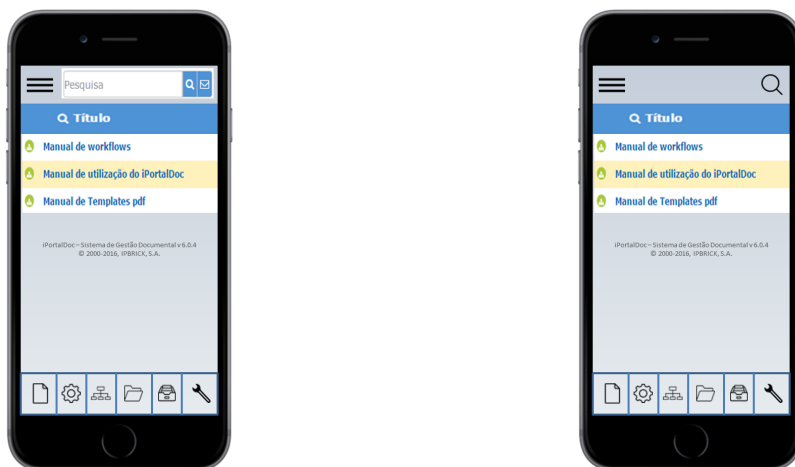


Figura 2.4: Possibilidades para a página frontal do iPortalDoc Mobile

Aqui, tendo por base a interface que será descrita no capítulo na subsecção 4.1.1, foram projetados seis ícones que substituem a barra de ferramentas da versão desktop - a rever tendo em conta os casos de uso a disponibilizar ao utilizador -, um ícone para mostrar e ocultar a hierarquia de documentos representado por três barras pretas horizontais e a zona onde se pode observar a lista de documentos. A principal diferença entre as duas possibilidades reside na zona de pesquisa, enquanto que uma permite a imediata introdução de texto, a segunda é um link para uma página onde são permitidas outras opções de pesquisa avançada ou mail archiving, representada na figura 2.5.



Figura 2.5: Possibilidade para a página frontal de pesquisa

Neste capítulo fizemos uma breve apresentação do problema que irá servir de base a esta dissertação, assim como uma foi apresentada uma solução empírica com mockups, apenas como ponto de partida.

O conjunto de modelos serviu para balizar a revisão bibliográfica sobre metodologias mais relevantes para a dissertação, bem como fornecer exemplos base para o desenvolvimento da solução final.

Capítulo 3

Revisão Bibliográfica

Este capítulo faz um levantamento dos trabalhos e artigos que abordam User Experience e User Interface no contexto de aplicações informáticas, assim como é produzido um estudo sobre as frameworks a utilizar para desenvolver o iPortalDoc Mobile recorrendo a linguagens de programação Web.

3.1 User eXperience (UX)

3.1.1 Definição

Uma dificuldade inerente ao conceito de User eXperience (UX) é a sua subjetividade e conseqüente dependência do indivíduo e do momento. Por exemplo, o que hoje pode ser uma experiência agradável amanhã poderá não ser. Aplicando esse conceito ao domínio empresarial, Dan Norman [10] engloba todos os aspetos da interação do utilizador final com os serviços e produtos de uma empresa. UX é uma consequência do estado interno do utilizador (as suas predisposições, expectativas, necessidades, motivações, humor, etc.), as características do sistema (a sua complexidade, propósito, usabilidade, funcionalidade, etc.) e o contexto ou ambiente dentro do qual a interação ocorre (situação organizacional/social, significado da interação, espontaneidade da utilização, etc.) [11]. M. Hassenzahl e N. Tractinsky [11] adiantaram ainda que, na perspetiva deles, um dos principais objetivos da interação computador-humano no futuro passará por contribuir para a qualidade de vida, projetando sempre em função do prazer em vez de simplesmente afastar sensações desagradáveis na utilização do sistema. Ou seja, a UX não está necessariamente ligada a aplicações mas sim à experiência que temos em relação ao mundo.

3.1.2 Decomposição

Embora UX esteja assente numa ambigüidade e abstratividade elevadas, é possível estruturar hierarquicamente o seu conceito para a tornar mais perceptível e mensurável. Foram feitos vários estudos neste sentido [5] [12] definindo os elementos e os sub-elementos da UX que são tidos em conta e que mostramos na tabela 3.1.

Tabela 3.1: Definição das dimensões hierárquicas da UX [5]

Elemento	Sub-elemento	Definição
Usabilidade	Simplicidade	Forma como um produto ou serviço funciona de maneira simples, sem complicações
	Sinceridade	Grau de percepção do utilizador em controlar a interface
	Eficiência	Grau de um produto ou serviço permitir a realização de uma tarefa sem perdas de tempo
	Informação	Grau de um produto ou serviço ao prestar as informações necessárias e adequadas ao utilizador
	Flexibilidade	Grau de um produto para receber alterações além das que foram especificadas inicialmente
	Aprendizagem	Tempo e esforço necessário para o utilizador aprender a usar o produto ou serviço
	Suporte	Habilidade para usar um produto ou serviço
Afeto	Cor	Grau em que a cor usada é agradável
	Delicadeza	Grau em que um produto é feito de forma fina e hábil
	Textura	Grau em que a textura ou o toque apela ao utilizador
	Luxúria	Grau em que o produto parece luxuoso, caro e de qualidade superior
	Atratividade	Percepção do utilizador de que o produto ou serviço é agradável, interessante e atraente
	Simplicidade	Forma de como um produto ou serviço é simples e descomplicado
Valor para o utilizador	Satisfação pessoal	Grau de satisfação que um produto ou serviço dá o utilizador consigo mesmo ou das suas conquistas
	Prazer	Sentimento de estar satisfeito devido à interação feita com o produto ou serviço
	Necessidade	Grau em que as funções ou aparência de um produto ou serviço satisfazem a necessidade do utilizador
	Sociabilidade	Grau em que um produto ou serviço satisfaz o desejo de ser sociável
	Ligação	Capacidade de atribuir valor subjetivo ao produto ou serviço
UX geral		Valores globais da experiência do utilizador ao interagir com o produto

3.1.3 Gamification

Relacionado com a UX está a Gamification, que se refere ao uso de elementos de design de um jogo em contextos diferentes de um jogo [13]. Pode ser definida como um processo de melhoria de um produto ou serviço partindo do estudo das experiências de jogabilidade, a fim de as replicar, acrescentando valor à experiência do utilizador [14]. O objetivo é a partir do entendimento da motivação que é aplicada na criação de um jogo, definir mecanismos que permitam motivar um utilizador na realização de determinada tarefa, que de outra forma poderia não ser cumprida ou ser entediante. Assim, as técnicas usadas passam por dar importância aos desejos, interesses pessoais ou sociais dos utilizadores do serviço. De entre as várias possibilidades de aplicação, uma das estratégias consiste na utilização de elementos gráficos, dos quais são exemplo: barras de progresso, recompensas, níveis ou até tabelas que classificam os vários utilizadores, gerando competição. Estes elementos podem ser apresentados de várias formas, sendo que existem três principais tipos de componentes a ter em conta na Gamification [15]:

- Visível - consiste no nome, na representação visual e na descrição do objetivo cumprido;
- Lógica - engloba os requisitos, o *trigger* e as condições para serem atingidas as metas;
- Recompensa - os prémios dentro ou fora da plataforma.

Desta forma, é tida em consideração a motivação natural das pessoas em socializar, aprender, dominar, competir e conquistar, usando-a para produzir comportamentos desejados na utilização de um serviço.

3.2 User Interface (UI)

3.2.1 Definição

Associado à UX, no contexto das aplicações, está a User Interface (UI) sem a qual não seria possível a interação homem-máquina. Pode ser entendida como uma junção de software e hardware que permite informar e receber comandos do utilizador. Como está explícito na publicação de Jobs, et al. [16] UI é a ponte de ligação através da qual os utilizadores recebem não apenas conteúdo mas também respostas a ações ou comportamentos, incluindo tentativas de aceder a recursos, ferramentas e funções de um dispositivo. Uma boa User Interface fornece uma experiência de utilizador agradável permitindo ao mesmo interagir com o software e hardware de uma maneira natural e intuitiva [17].

3.2.2 Princípios

Existem vários princípios gerais que constituem boas práticas para a UI de aplicações Web. Segue-se uma lista de princípios gerais com base no trabalho de Blair-Early e Zender [9].

- Ponto de partida - Para um utilizador perceber a interface é importante que esteja bem definido o local por onde deve começar. Características como tamanho, movimento ou cor devem ser consideradas quando se quer mostrar ao utilizador o que deve fazer. Fazendo uma analogia a uma casa, é importante mostrar a posição e a funcionalidade da porta de entrada óbvia.
- Saídas claras - O utilizador deve saber por onde deve sair ou cancelar uma ação. Tal como no ponto de partida deve ser óbvio como se pode reverter alguma situação. A Apple chama este princípio "perdão", as pessoas precisam de se sentir confortáveis sabendo que podem efetuar ações sem danificar o sistema.
- Lógica consistente - Numa interface deve ser possível ao utilizador perceber rapidamente a lógica do sistema e os padrões, que consistente e racionalmente ligados a conteúdos permitem o fácil reconhecimento desses mesmos padrões.
- Convenções familiares de interface - É importante respeitar experiências sociais culturais anteriores dos utilizadores. Estes chegam às interfaces com um background de experiências e expectativas em relação a linguagem ou imagens e não com uma página em branco.
- Feedback - Para saber que as suas ações estão a ter efeito, é necessário dar um feedback imediato para manter os utilizadores informados. Quanto mais personalizado à ação mais interessante pode ser, é uma forma de recompensar o utilizador e mantê-lo o interesse na aplicação.
- Referência - Os utilizadores devem ter conhecimento do local onde se encontram de uma forma que seja fácil de identificar.

- Proximidade - Não faz sentido para um utilizador que seja difícil realizar ações ou aceder a conteúdo semelhantes. Existem pelo menos três tipos de proximidade: a de espaço que acontece quando existe uma lógica de evolução na associação conteúdo-interface, a temporal que está relacionada com ter o conteúdo disponível quando o utilizador pretende e a de conteúdo que consiste em agrupar itens semelhantes.
- Adaptabilidade - É importante desenhar interfaces que se adaptam às necessidades dos utilizadores, por exemplo ocultar menus que não são usados frequentemente ou dar mesmo essa possibilidade de escolha.
- Ajuda - É importante ter disponível um manual de utilizador. Este deve estar disponível mas deve ser apresentado de forma subtil.
- Importância do conteúdo - Um utilizador recorre a uma interface para aceder a conteúdo logo este tem que ser primordial. A interface é parte desse conteúdo e essa interação tem que ser o mais direta possível.

A lista estudada deve ser considerada para qualquer aplicação Web mas no contexto específico da interface móvel é importante ter-se em consideração também outros aspetos. Existem diferenças entre utilizar um telemóvel e um computador quando se quer aceder a uma aplicação porque os objetivos e o tempo disponível são distintos. A finalidade na utilização de um telemóvel consiste em executar algo elementar e célere. Assim surgem alguns princípios completamente focados na interface móvel que permitem uma boa interação com o utilizador presentes no artigo escrito pelo Creative Bloq Staff e baseado nos Workshops de Jonathan Stark [18]:

- Mentalidade - A mentalidade difere obrigatoriamente quando o objeto de estudo deixa de ser o ecrã de computador e passa a ser o de um dispositivo móvel. Este último, sendo de tamanho reduzido, obriga a que se tenha presente o importante princípio que algo vai ter que ser deixado de fora porque nem tudo cabe no ecrã. Por outro lado, algo que não pode ser excluído é a experiência do utilizador: não basta apenas pensar no que seria interessante desenvolver do ponto de vista do programador mas também do utilizador para que seja possível atingir as metas estabelecidas para a aplicação.
- Contexto - A utilização de um smartphone num contexto de trabalho, em frente a uma secretária, é invulgar. Muitas pessoas utilizam-no, por exemplo, sentadas no sofá interrompendo repetidamente o que estão a fazer para abrir um novo separador no browser. Existem também momentos em que se usa o telemóvel quando nos estamos a mover de um sítio para outro o que faz com que seja fundamental existir na aplicação, a capacidade de realizar pequenas tarefas rapidamente. Aqui entra também em conta o fator da visão que, por estarmos em andamento, é reduzida e não estamos totalmente concentrados, o que faz com que ícones grandes ou cores desempenhem um papel importante. Por forma a facilitar o trabalho do

utilizador é interessante que no momento em que se volta a abrir uma aplicação esta continue no ponto em que foi deixada. O objetivo é tornar a experiência do utilizador intensa e inesquecível.

- Linhas gerais
 - Respostas - No momento em que o utilizador interage com a aplicação, a mesma tem que apresentar resposta a essa interação, mesmo que seja apenas com objetivo de informar que o pedido está a ser processado e é necessário aguardar.
 - Padrão e metas - Com a invenção e a aposta nos touchscreens, o rato do computador foi substituído pela mão do utilizador, mais concretamente o seu polegar. Isto faz com que seja importante projetar uma aplicação partindo do princípio que por vezes se usam as duas mãos mas que maioritariamente das vezes o utilizador usa apenas uma, fazendo com que diferentes locais do ecrã sejam alcançados de maneiras distintas. É difícil tocar em pontos pequenos do ecrã portanto, exceções à parte, uma boa regra para o tamanho de um elemento de uma interface mobile é ter 44 pixels.
 - Conteúdo - O conteúdo tem que ser o foco. Os touchscreens permitem uma interação direta entre o utilizador e as aplicações assim, todas as distrações e pontos irrelevantes devem ser colocadas de parte e dar prioridade ao conteúdo na frente e centrado ao ecrã. O scroll também deve ser evitado, salvo em algumas exceções, pois torna a aplicação menos previsível.
 - Controlos - Ao contrário dos ecrãs de computador, deve optar-se por colocar controlos, como ações para mudar de aba, na parte inferior do ecrã para que rapidamente se veja o resultado da ação. Caso contrário, se os controlos estiverem na parte superior, parte da mão retira a visibilidade no momento da interação.
- Modelos de Navegação - Existem três tipos de navegação mais comuns: nenhum, que são aplicações de ecrã único, barra de separadores, que permitem várias áreas de conteúdo e listas, que mostram conteúdo hierarquicamente quando selecionadas.
- Inputs - É importante adequar a forma para o utilizador introduzir dados a esses mesmos dados (texto, número, etc.)
- Orientação - Para aplicações que necessitem da introdução recorrente de dados pelo utilizador pode fazer sentido disponibilizar a versão de ecrã em landscape para assim terem acesso a um teclado maior.
- Comunicação - Um dos princípios mais importantes passa por dar feedback ao utilizador, seja este sensível (vibração) ou visual (destacar texto). Numa ação mais demorada ou complexa pode fazer sentido apresentar uma barra de progresso ou um diálogo de confirmação (por exemplo, "De certeza que quer eliminar o documento?"). Os alertas são importantes mas não devem ser usados de forma excessiva pois são extremamente intrusivos e podem

ser desagradáveis para o utilizador, devendo apenas ser utilizados em situações onde algo de errado aconteceu.

3.2.3 Alternativas

Numa perspetiva mais técnica importa referir que para o caso concreto de aplicações móveis existe a possibilidade de criar uma aplicação nativa escrita, tipicamente, em Android ou em Objective C/Swift para iOS, ou fazer uma aplicação híbrida com o website embebido numa aplicação nativa usando uma framework de desenvolvimento e uma de interface de aplicações móveis, recorrendo a linguagens de programação Web, HTML, PHP, CSS, JavaScript e AJAX. No caso desta dissertação, a solução escolhida e proposta pela IPBrick foi a mencionada em segundo mas foram estudadas as vantagens e desvantagens de ambas. A primeira opção tem vantagens a nível de gráficos e animações, que é aconselhada quando se precisa de muitas aplicações gráficas, como num jogo. Em contrapartida não permite portabilidade para outro tipo de plataformas – estaria limitada à plataforma para a qual fosse desenvolvida, por exemplo Android ou iOS -, tem um custo elevado de desenvolvimento, de manutenção e leva muito tempo a ser desenvolvida. Já a segunda solução tem as desvantagens de ter gráficos limitados, exige a familiarização com mobile frameworks mas oferece as vantagens seguintes: assemelha-se muito a uma aplicação nativa sem todo o custo envolvido, permite a utilização de tecnologias Web, com reutilização de código, assim como a integração mobile. Permite o acesso à câmara, microfone ou lista de contactos. Esta solução é vantajosa porque possibilita que apenas a mesma aplicação consiga atingir e funcionar na maioria das plataformas de smartphone [19].

3.3 Frameworks

Uma framework consiste numa abstração que permite programas comuns utilizarem uma funcionalidade genérica já desenvolvida, acelerando assim o processo de desenvolvimento de código de uma forma estruturada, permitir escalabilidade nos projetos e garantir condições de segurança. Uma framework oferece soluções para problemas semelhantes, usando um conjunto de classes e interfaces flexíveis e extensíveis que permitem construir várias aplicações com pouco esforço, especificando apenas as particularidades de cada aplicação. [20]

3.3.1 Frameworks de desenvolvimento de aplicações móveis

Como já foi mencionado, a finalidade do projeto é criar uma aplicação móvel que funcione em todas as plataformas sem a necessidade de recorrer às linguagens específicas de cada dispositivo devido aos seus custos de desenvolvimento e incompatibilidade multi-plataformas. Para combater estas desvantagens a solução é a utilizar a Web. Esta é a única plataforma que faz com as aplicações sejam acessíveis a partir de todos os dispositivos através de um browser e, embora a sua utilização tenha sido reduzida durante muitos anos para aplicações móveis, a procura por produtos

e serviços sem necessidade do controlo de operações simultaneamente com o desejo de compatibilidade nas diferentes plataformas tornou a Web uma das plataformas de aplicações com um maior crescimento até à data [21]. Para permitir que as tecnologias Web funcionem como uma aplicação e que não haja um desperdício de tempo são necessárias ferramentas de trabalho/frameworks. A finalidade de uma framework de desenvolvimento de aplicações móveis é fazer com que as aplicações escritas em HTML, JavaScript e CSS sejam executadas em formato WebView para ser possível serem acedidas através de um dispositivo móvel. Concetualmente vai colocar a aplicação Web empacotada dentro de uma aplicação nativa, onde o JavaScript tem acesso às APIs do dispositivo. Existem diferentes alternativas neste contexto, com as próprias vantagens e desvantagens [22].

3.3.1.1 Apache Cordova/PhoneGap

O Apache Cordova, também conhecido PhoneGap é a principal framework na partilha de conhecimento entre programadores. Tem a vantagem de permitir aos programadores que já estão familiarizados com as ferramentas Web de imediatamente aumentar as suas capacidades, pois podem construir uma aplicação mobile sem a necessidade de conhecer as linguagens específicas de cada plataforma. A instalação de uma aplicação Cordova é exatamente igual a uma aplicação nativa. É open-source e grátis, o que não traz custos financeiros de desenvolvimento mas também faz com que o uso da plataforma não seja uma garantia de sucesso. É necessário encontrar os plugins necessários, ou construí-los, para utilizar características nativas especiais. A performance do Cordova pode por vezes não ser a melhor portanto é importante utilizar frameworks de UI adaptadas a mobile.

3.3.1.2 Appcelerator

A Appcelerator proporciona uma API de JavaScript compatível com todas as plataformas, com as características específicas das mesmas. Permite o uso das componentes de UI nativas com boa performance relativamente a outras frameworks. O código da aplicação é normalizado com JavaScript o que permite o desenvolvimento de capacidades para quem programa. Tem a desvantagem de tornar a compilação difícil para várias plataformas e é preciso estar treinado para conseguir transferir a tecnologia para o exterior.

3.3.1.3 Adobe AIR

O Adobe AIR um sistema operativo multi-plataforma que permite aos programadores combinar HTML, JavaScript e Adobe Flash. Tem a desvantagem de não permitir o uso de HTML e JavaScript para escrever aplicações mobile e o fato da Adobe ter comprado os direitos do nome PhoneGap dá a entender que a AIR poderá não estar no mercado durante muito mais tempo.

3.3.1.4 Sensha

A Sensha criou diversos produtos para tornar o desenvolvimento de aplicações mais facilitado. Um construtor de aplicações visual para HTML5, um para visualização de gráficos, entre outros. Oferece bibliotecas para componentes de UI, uma API extensível e templates já construídos. Sofre os mesmos problemas das aplicações Cordova se os programadores não forem eficientes no desenvolvimento de código JavaScript, escrito com Sensha Touch.

3.3.1.5 Qt

Cobre diferentes plataformas mobile, fornece um número substancial de bibliotecas, com APIs intuitivas, tem ferramentas de desenvolvimento sólido e dá suporte em diferentes linguagens. Tem custos financeiros que ultrapassam os 4000 € e não é open-source o que não permite o desenvolvimento de capacidades de quem desenvolve a aplicação.

3.3.2 Frameworks de User Interface de aplicações móveis

As frameworks para UI de aplicações móveis permitem uma forma mais rápida de utilizar os componentes de UI adaptados a dispositivos de comunicação mobile em vez da criação de raiz destes elementos, o que levaria muito mais tempo para desenvolver o projeto. Ferramentas como scroll, toggle, tabs, accordion, forms, dropdown ou swipe já foram concebidas e o seu código perfeitamente desenvolvido e estruturado, o que permite a sua utilização invocando simplesmente as funções da framework. Neste contexto, são apresentadas várias alternativas no mercado e descritas as suas características principais.

3.3.2.1 Ionic

- Grátis;
- Open Source;
- Integrada com Apache Cordova;
- Bibliotecas otimizadas e inclui HTML 5;
- Boa documentação;
- Comunidade em crescimento e fóruns bastante ativos.

3.3.2.2 F7

- Grátis;
- Open Source;
- Não é compatível com todas as plataformas;

- Componentes otimizados para iOS;
- Excelente documentação;
- Fóruns pouco ativos.

3.3.2.3 Mobile Angular UI

- Grátis;
- Open Source
- Compatibilidade por vezes difícil;
- Fornece componentes que estão em falta no bootstrap [23];
- Documentação superficial e vaga em partes;
- Fóruns pouco ativos.

3.3.2.4 Kendo UI

- Preço elevado;
- Não é Open Source;
- Componentes utilizados podem variar consoante as plataformas;
- Fornece os últimos componentes HTML, CSS e JS;
- Pouca documentação;
- Fóruns pouco ativos.

3.3.2.5 Sensa Touch

- Preço elevado;
- Open Source;
- Não é garantida compatibilidade com todas as plataformas;
- Permite a criação de aplicações completas e complexas;
- Excelente documentação;
- Fóruns ativos.

3.3.3 Frameworks PHP

Para permitir a comunicação com a base de dados é obrigatório utilizar uma linguagem que seja reconhecida e possível de ser interpretada pela mesma, por isso optou-se pela já conhecida e usada PHP. De forma a diminuir o tempo de desenvolvimento resolveu-se conhecer e estudar as frameworks desta linguagem com base no documento fornecido pela IPBrick da formação que existiu sobre o mesmo tema. Em [24] enumera-se à seguinte lista de vantagens e desvantagens das frameworks de PHP mais utilizadas:

3.3.3.1 Laravel

Vantagens:

- Organização de ficheiros e código;
- Rápido desenvolvimento aplicacional;
- Arquitetura MVC e PHP 7;
- Testes unitários (Rápidos no HHVM);
- Conhecido como tendo a melhor documentação;
- Alto nível de abstração;
- Capaz de Overloading, usando métodos dinâmicos;
- Várias funcionalidades extra (Fora da caixa);
- Integração de meios de pagamentos c/STRIPE;
- Pacotes com criptografia bastante segura;
- ORM.

Desvantagens:

- Não funciona em alojamento partilhado;
- Faz muitas consultas à Base de dados.

3.3.3.2 Symphony

Vantagens:

- Bom desempenho, devido ao cache byte code;
- Estável;
- Bem documentado, mantido e suportado;

- Bom suporte e muito maduro;

Desvantagens:

- Curva de aprendizagem íngreme;
- As empresas estão a mudar para arquitetura MVC, Symfony2 não suporta MVC.

3.3.3.3 CodeIgniter

Vantagens:

- Desenvolvimento amigável. Não necessita de dependências especiais ou suporte;
- Facilidade no uso de serviços de alojamento e usando BD's padrão como MySQL;
- Fácil integração, suporta a maioria de outras frameworks que não utilizem o MVC;
- Boa documentação e suporte de longo prazo.

Desvantagens:

- Não usa namespace's (Por sua vez acelera o sistema);
- É menos amigável para testes;
- Poucas bibliotecas que são construídas dentro do framework.

3.3.3.4 CakePHP

Vantagens:

- Estrutura moderna, suporta versão superior a 5.5;
- Desenvolvimento rápido;
- Bom para aplicações Web comerciais (Licença MIT);
- Acesso a BD, Cache, Validações, Autenticação;
- Extensas ferramentas de segurança inclui cross-site;
- Prevenção de Script, Prevenção de SQL injection;
- CSRF e validação de formulários;
- Boa documentação;
- Desenvolvimento ativo.

Desvantagens:

- Não é tão indicado para a construção de RESTFUL APIs como Laravel ou outros frameworks.

3.3.3.5 Zend

Vantagens:

- Ideal para aplicações empresariais;
- Orientado a objetos;
- Muitos componentens para validação, feeds e formulários;
- Desacoplado.

Desvantagens:

- Não favorece o desenvolvimento rápido de aplicações.

3.3.3.6 Yii2

Vantagens:

- Rescrito do Yii1;
- Moderno e flexível;
- Um das mais antigas frameworks (Yii1) e ainda suportada;
- Tem pacotes para autenticação e segurança;
- Tempo curto de desenvolvimento e aprendizagem;
- Muito configurável

Desvantagens:

- Se não comentar o código, este rapidamente pode ficar confuso

3.4 Conclusão

Neste capítulo apresentámos conceitos relacionados com a interface de utilizador de aplicações informáticas, nomeadamente User Experience e User Interface e ainda os princípios de os princípios de Gamification, Lógica consistente, Feedback, Importância do conteúdo, Mentalidade e Controlos. O capítulo também apresenta um levantamento de tecnologias relevantes para o desenvolvimento de uma aplicação móvel. Nos capítulos seguintes iremos usar estes conceitos, princípios orientadores e tecnologias para produzir uma aplicação móvel, nomeadamente o iPortalDoc Mobile, simples, intuitiva, inteligível e aprazível para o utilizador.

Capítulo 4

A plataforma iPortalDoc

Este capítulo contextualiza e pormenoriza a plataforma sobre a qual vai incidir a dissertação, nomeadamente a plataforma iPortalDoc da IPBrick, abordando detalhadamente a interface e os casos de uso respetivos. A partir desta análise o capítulo apresenta o modelo e requisitos a considerar para o desenvolvimento da versão móvel iPortalDoc Mobile.

4.1 iPortalDoc

O iPortalDoc é a aplicação web de um sistema de gestão documental e Workflow da IPBrick que permite aos seus utilizadores gerir o fluxo dos documentos de uma empresa. Possibilita o registo e o tratamento de todas as comunicações como e-mails, conversas de chat, chamadas telefónicas, assim como a utilização de interfaces de listas de documentos, pesquisas e gestão de e-mails [8].

O iPortalDoc serve como repositório e gere fluxos de trabalho com recurso a workflows, constituídos por várias etapas às quais estão associadas ações a realizar por um ou vários utilizadores e por transições com condições lógicas que permitem transitar para o estado seguinte. O iPortalDoc Mobile foi idealizado para possibilitar ao utilizador realizar as suas ações de vários workflows. As ações, representadas por um formulário, estão relacionadas com o dia a dia do trabalho das empresas e podem ser de vários tipos, como por exemplo conferir fatura, classificar carta e distribuir ou inserir orçamento e enviar proposta. O formulário pode variar e ter elementos como campo para comentário, lista com informação dos documentos associados, campos para preencher e-mail e lista de utilizadores ou grupos a selecionar para encaminhar. Todos os constituintes da ação estão definidos na base dados respetiva. Para executar ações podem existir restrições dependendo do que foi definido previamente, como a obrigatoriedade de preencher campos relativos a comentários ou consultar o esquema cronológico do documento associado à ação, e depois é necessário pressionar um dos botões de encaminhamento, selecionando um utilizador ou vários, ou de execução que, para o tipo conferir fatura, podem ser conforme, não conforme ou para nova conferência.

4.1.1 Interface

O acesso pela interface Web para gestão de documentos é dividido em quatro áreas: hierarquia de documentos, documentos, barra de ferramentas e pesquisa.

- Hierarquia de Documentos - Os documentos são inseridos dentro de pastas correspondentes ao seu Workflow. Cada pasta é constituída por sub-pastas que correspondem ao ano, mês e, em alguns casos, ao dia do documento.

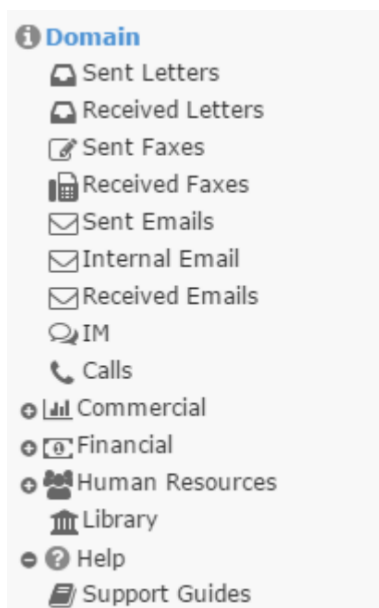
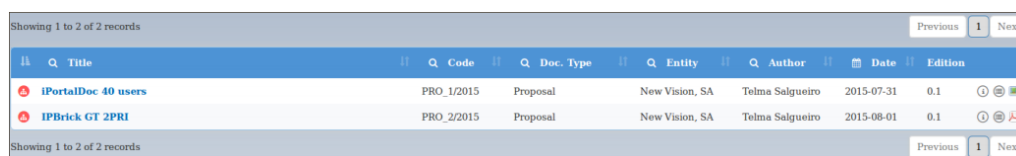


Figura 4.1: Hierarquia de Documentos

- Documentos - Na zona central da página está a área de documentos onde é visualizada a lista de documentos. Na Fig. 4.2 é apresentada uma lista de documentos selecionada pelo utilizador na hierarquia.



Q	Title	Q	Code	Q	Doc. Type	Q	Entity	Q	Author	Q	Date	Q	Edition	
	iPortalDoc 40 users		PRO_1/2015		Proposal		New Vision, SA		Telma Salgueiro		2015-07-31		0.1	⊕ ⊖ ⌂
	IPBrick GT 2PRI		PRO_2/2015		Proposal		New Vision, SA		Telma Salgueiro		2015-08-01		0.1	⊕ ⊖ ⌂

Figura 4.2: Documentos

- Barra de Ferramentas - Como mostra a Fig. 4.3 é possível aceder aos separadores *Document*, *Definitions*, *Workflow*, *Folder*, *Modules* e *Session*. Em *Document* pode-se visualizar informação e executar ações sobre documentos. *Definitions* permite aceder a definições do sistema sobre perfis, utilizadores e grupos, assim como criar novos tipos de documentos e templates. Para manipular fluxos de documentos recorre-se ao separador *Workflow* e para associa-los aos utilizadores recorre-se a *Folder*. Se se quiser aceder a estatísticas ou configurar a sessão atual usa-se os separadores *Modules* e *Session*, respetivamente.



Figura 4.3: Barra de Ferramentas

- Pesquisa - Forma rápida de encontrar documentos sem ter que explorar todas as diretorias. A pesquisa, por defeito, é feita em todas as pastas e sub-pastas recursivamente. Existe a possibilidade de se fazer uma pesquisa rápida ou avançada.



Figura 4.4: Pesquisa

4.1.2 Casos de Uso

O iPortalDoc possibilita diversas interações com os seus utilizadores sejam relacionadas com pesquisa, documentos ou definições. A partir do Manual de Utilizador [25], procedeu-se ao levantamento de todas essas possibilidades. Na Fig.4.5 está a lista de todos os casos de uso.

Search		Definitions		Workflow
Quick search		Configuration Interface		Create
Advanced search		iPortalDoc Logo		Instantiate
Documents		iPortalDoc Language		Configure
Navigation area in document hierarchy		Active Modules		Remove instance
Documents visualization area		Project Management Module		Change T.F
Without Selecting a Document		DBDOCMAILS Module		View T.F
Insert		Stamper		Exec Time
Remove		Document Data		Doc Type Assoc
Link Docs		Workflow Data		Assoc Folder
Move Docs		Actions to perform		Actions Generator
Selecting a Document		Associated Emails		Folder
Actions		Validation Expiration		Change
Open		Digital Signature		Create
Insert		Change License		Remove
Forward		Pop-List Menu		Move
Add commento		Additional modules		Customize
Remove		Aux. Fields		Permissions Management
Update Rev (new edition)		Print		Replicate permissions
Edit info		Calendar		Info
Associate		Interfaces		List of Actions
Dissociate		Actions		Actions by user
Link		Workflows		Modules
Link Documents		Users		Statistics
Tag		Graphic Layout		Session
Move		Release Notes		Actions
Move Docs	Users			Calendar
Permissions	Profile			Entities
Print	Aux Fields			Order Docs
Cancel Workflow	Assoc User			Config User
Information	Doc Types			Absences
Doc info	Templates			+info Management
Assoc Doc	Macros			Predefined comments
Assoc Mais	Build Global File System			Change Acronym
Editions	Build Global Mail File System			History
Doc Workflow	iPortalDoc Light			Graphic Layout
Doc Actions	Projects			Build Filesystem
				Logout

Figura 4.5: iPortalDoc - Casos de Uso

Os casos de uso na secção *Folder*, relativos às opções de *open*, *forward*, *add comment*, *tag* e *doc info* na secção *Documents* e ainda os casos de *absences* e *predefined comments* na *Session* são passíveis de serem transitados para a versão móvel da aplicação.

Este conjunto de interfaces e casos de uso da aplicação desktop é particularmente relevante para se estruturar os objetivos para a versão móvel.

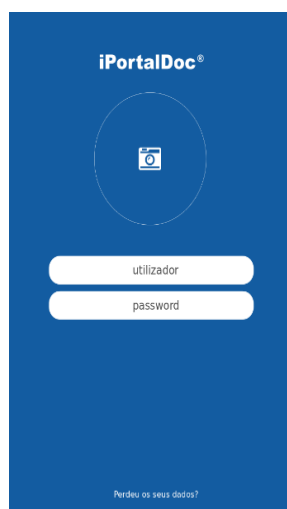
4.2 Modelo e Requisitos do iPortalDoc Mobile

Com base nas funcionalidades da aplicação original iPortalDoc e considerando os pedidos específicos do cliente, os designers da IPBrick propuseram um modelo inicial para o layout da versão mobile que apresentamos nesta secção. Apresentamos também os requisitos técnicos a que a aplicação iPortalDoc Mobile deverá obedecer, que foram fundamentais para definir metas e objetivos de layout e funcionamento da aplicação.

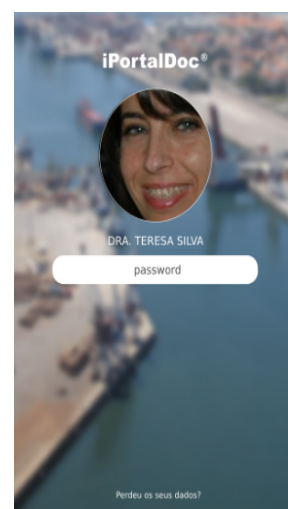
4.2.1 Login

Requisitos técnicos:

- Tempo máximo para login de 3 segundos
- Guardar dados de login
- Possibilidade de personalizar imagem



(a) Sem informação de utilizador



(b) Com informação de utilizador

Figura 4.6: Modelos para a página Login

4.2.2 Menu

Requisitos técnicos:

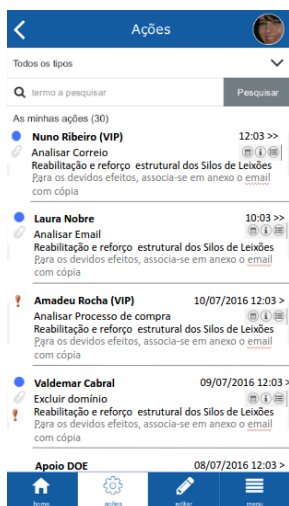
- Personalizar dados pessoais
- Botão no canto superior direito para aceder às opções de menu
- Poder personalizar alertas e notificações
- Permitir personalizar lista de utilizadores VIP
- Possibilidade de inserir comentários pré-definidos
- Poder inserir ausências

4.2.3 Lista de ações

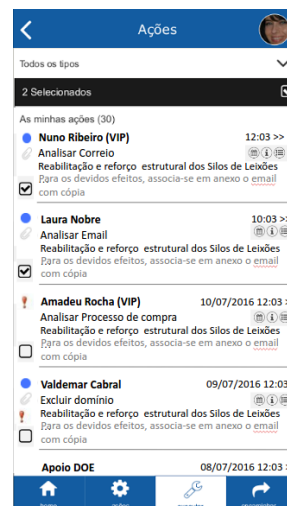
Requisitos técnicos:

- Filtrar por tipo de Ação
- Pesquisar por ações
- Ter acesso à informação do número de ações
- Botões de acesso rápido azuis no footer que evidenciam seleção alterando a sua cor para branco
- Informação com o título da ação a azul e estado do workflow a vermelho ou verde
- Ícones à direita da ação para permitir agendar, aceder à informação ou ao esquema cronológico
- Marcar ação como não lida deslizado a mesma para a direita
- Deslizar ação para a esquerda e ter possibilidade marcar ou ter mais opções
- Sinalizadores de importância
- Ícone para informar se a ação tem documentos associados e outro para o caso da ação ter vários encaminhamentos
- Duas linha para o comentário da ação
- Tamanhos iguais, em altura e largura
- Um toque na ação permite entrar e um toque continuado permite copiar texto
- Executar em bloco

- Possibilidade de selecionar mais do que uma ação para executar em bloco. Ao selecionar os botões de acesso rápido no footer mudam para permitir executar ou encaminhar ações mais rapidamente
- Confirmação para executar ou encaminhar ações



(a) Modelo para a página Lista de ações



(b) Modelo para a página Lista de ações com seleção

Figura 4.7: Modelos para a página Lista de ações

4.2.4 Ação

Requisitos técnicos:

- Caixa de texto para comentário com a possibilidade de guardar e cancelar
- Dar acesso aos comentários por grupos
- Filtrar lista de utilizadores por grupos
- Visualizar notas e documentos associados
- Campo de pesquisa para documentos associados
- Botões de resultados e sub-ações por menu drop down
- Botões inferiores de info documento / esquema cronológico / agendamento / encaminhamento
- Possibilidade de escolher comentários pré-definidos
- Input para inserir endereço de e-mail a enviar por CC
- Permitir copy-paste

- Layout de acordo com o novo iPortalDoc
- Título da ação / código / entidade / data
- Campo assunto e mensagem
- Campos para notificação / email / tempo limite
- Encaminhamentos

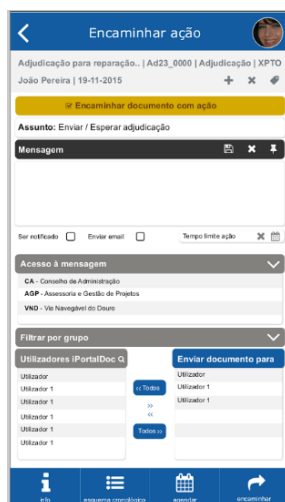


Figura 4.8: Modelo para a página Ação

4.2.5 Informação da Ação

Requisitos técnicos:

- Informação com título do documento, código e entidade
- Lista com dados do documento
- Acesso ao esquema cronológico e agendamento

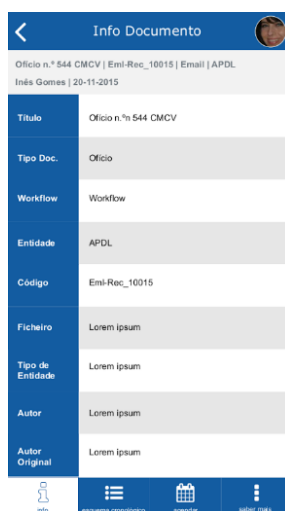


Figura 4.9: Modelo para a página Info

4.2.6 Esquema Cronológico da Ação

Requisitos técnicos:

- Dropdown de menu de utilizadores
- Iconologia para fácil leitura e interpretação dos dados
- Permissão para escolher forma como a informação é mostrada (tabular, resumida ou detalhada)

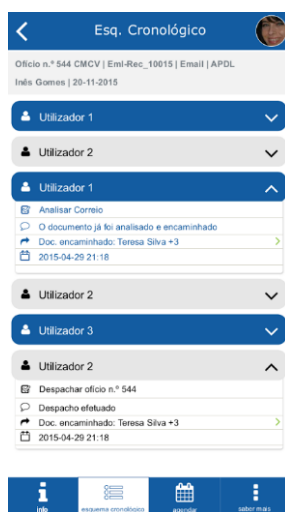


Figura 4.10: Modelo para a página de Esquema cronológico

4.3 Conclusão

Neste capítulo descrevemos a plataforma iPortalDoc a partir da qual foram desenvolvidos os modelos e definidos os requisitos técnicos para a versão iPortalDoc Mobile. Os modelos e requisitos aliados à interface e casos de uso analisados serviram para definir em concreto quais as áreas de foco da versão mobile, permitiram estabelecer o plano de trabalho sobre as tarefas que a aplicação mobile vai exigir, assim como fornecer objetivos e metas para o desenvolvimento da solução final.

Capítulo 5

A solução iPortalDoc Mobile

Este capítulo descreve a solução desenvolvida para a aplicação móvel iPortalDoc Mobile, começando pela estruturação das partes de frontend e backend da aplicação, seguindo-se a definição das frameworks a utilizar, a arquitetura de funcionamento da aplicação com as classes, métodos e variáveis, e termina com uma referência à implementação da mesma.

5.1 Estrutura

5.1.1 Frontend

O frontend diz respeito a tudo o que o utilizador vê, estando intrinsecamente ligado à UX e UI [26]. O objetivo da aplicação consiste na consulta e realização de ações pelo utilizador, portanto é obrigatório disponibilizar esse tipo de interação. Tendo em conta os requisitos definidos foram projetadas as páginas de login, home, menu, lista de ações, ação, informação, agendamento, esquema cronológico e marcação. A página home permite navegar para a lista de ações de forma integral, filtrada por critérios já existentes ou por outros que podem ser criados em qualquer altura. A página de menu permite efetuar configurações de utilizador, que por sua vez contém três páginas Lista VIP, Comentários Pré-definidos e Lista de Ausências, sendo que as duas últimas fornecem ainda a possibilidade de aceder a uma página individual de comentário e ausência respetivamente. As páginas de informação e esquema cronológico são as únicas da aplicação que permitem apenas consultar informação sobre as ações ou documentos.

Para o layout e design da aplicação, o modelo foi definido em 4.2, o que fez com o trabalho consistisse em aproximar esse mesmo modelo tanto quanto possível e também em propor melhorias estudadas no contexto de UX e UI que não estão presentes:

- Loading - A aplicação pressupõe desde logo comunicações com o servidor o que, mesmo com o desempenho otimizado vão demorar tempo, o sinal de ligação à internet pode não ser forte e assim é importante dar feedback ao utilizador que estão a ser efetuadas operações em background para que este perceba que vai ter que esperar para ter a possibilidade de continuar a trabalhar;

- Respostas - Ainda na lógica de feedback foram projetadas respostas ao utilizador sempre que é efetuada alguma operação, como mensagens de sucesso ou de erro relativamente à execução de ações para que não haja dúvidas sobre o que acabou de acontecer;
- Pesquisas - O modelo já continha uma barra de pesquisa na listagem de ações mas projetaram-se outras para a pesquisa de documentos, seleção de utilizadores e grupos. As listas de seleção podem ter um número reduzido ou atingir as dezenas portanto definiu-se que quando não é possível visualizar a lista na sua totalidade sem a necessidade de fazer scroll coloca-se uma barra de pesquisa com o intuito de proporcionar uma melhor experiência ao utilizador;
- Tamanho dos ícones de acesso - Na listagem de ações são projetadas várias formas de interação: apenas numa linha podemos selecionar a ação, clicar no ícone de informação, esquema cronológico e agendamento. Baseado no estudo feito vai-se garantir que os ícones de acesso tenham no mínimo 44 pixels para que não seja difícil ao utilizador selecionar aquilo que pretende;
- Localização dos botões de execução - Foram projetados os botões para a parte inferior do ecrã porque, como estudado, em regra geral o utilizador comum está sempre com o polegar disponível e próximo dessa zona, o que faz desse local, em vez do superior, o mais adequado para um acesso rápido.
- Localização das mensagens - Seguindo a lógica do ponto anterior, o polegar está sempre na parte inferior do ecrã e por vezes até pode impedir uma visão clara dessa mesma zona, assim a zona superior torna-se mais livre para visualizar mensagens.
- Convenções familiares de interface - Com base nos conceitos de UX estudados surgiu a ideia de projetar um ícone que intuitivamente sugira uma mudança no contexto específico do iPortalDoc de trocar lista de utilizadores para lista de grupos, em vez do ícone atual que mostra uma ou mais pessoas.
- Proximidade - Com apoio no conceito estudado relativo à proximidade, para minimizar o trabalho ao alternar entre conteúdo semelhante, projetou-se na página de ação dois ícones (uma seta voltada para cima e outra para baixo) para dar a possibilidade de navegação entre ações, sem a necessidade de voltar à página de listagem.
- Adaptabilidade - Neste contexto, pensou-se fazer uma interface que se adapte às necessidades do utilizador do iPortalDoc Mobile. Na página de menu está prevista a possibilidade de criar, editar e eliminar filtros de pesquisa de ações para que evitar que a aplicação seja estática e permita uma personalização de forma a satisfazer as necessidades de cada utilizador.

5.1.2 Backend

Numa comunicação cliente-servidor como é o caso da plataforma em que incide este projeto, o backend define-se como o lado do servidor, é responsável pela mudança de conteúdo e refere-se a temas como segurança e base de dados [26]. O iPortalDoc Mobile foi desenhado com o objetivo de ser uma aplicação com conteúdo dinâmico, o que obriga a que tenha acesso a uma base de dados capaz de armazenar informações de utilizador e fornece-las a qualquer momento. Utilizou-se a base de dados do iPortalDoc para que o conteúdo que se vê na aplicação móvel fosse igual à de desktop, da mesma forma que acontece quando consultamos por exemplo uma conta de e-mail no computador e no smartphone o conteúdo é o mesmo, muda apenas o frontend. Por si só a base de dados não interage com a aplicação portanto é indispensável existir algo que possibilite esta comunicação e para isso foi projetada uma *API*, que pode ser entendida como o servidor da aplicação, que comunica com a base de dados e retorna informação. A *API* é também responsável por tratamento de erros nomeadamente definir o modelo e as regras dos parâmetros que pode receber, com ajuda da framework PHP pode estabelecer-se que, por exemplo, se vai receber um parâmetro chamado nome e que tem que ser uma string com no mínimo 0 e no máximo 50 caracteres, caso contrário a comunicação é rejeitada. Esta camada ao estar separada de tudo o resto permite uma maior escalabilidade do projeto no futuro ao permitir a fácil inserção de novos controladores. De forma a tornar mais independentes as camadas de frontend e backend foi criada uma classe *Services* responsável por enviar e retornar os pedidos ao servidor e da *Local Storage*, ou armazenamento local, como o caso das *cookies*. Os *Services* foram também projetados para serem responsáveis pelo tratamento de erros HTTP, estando previstos alguns, como por exemplo, os erros 401 e 403 que correspondem a um pedido legítimo mas que o servidor se recusou a responder porque a autenticação, embora possível, não foi feita ou não foi fornecida e a um pedido legal mas que o servidor simplesmente se recusou a responder [27]. Com a descrição das classes *API* e *Services* ficou assim estruturado o backend do iPortalDoc Mobile.

5.2 Escolha das Frameworks

Com base no levantamento apresentado nas secções 3.3.1 e 3.3.2 chegou-se à conclusão que as melhores frameworks para utilizar neste projeto são a Apache Cordova, Ionic e Yii. As razões que levaram à escolha das frameworks de frontend são as seguintes:

- Grátis e Open Source - O projeto não tem orçamento, o que significa que os gastos devem apenas ser no desenvolvimento da aplicação e não na aquisição de ferramentas de trabalho. Não existe espaço para pensar sequer na possibilidade de se usar uma ferramenta que não seja grátis. O fato de ser open source possibilita o desenvolvimento de capacidades de programação.
- Compatibilidade Ionic-Cordova - As ferramentas da Framework Ionic utilizam Cordova nas suas camadas inferiores e este é um argumento robusto para evitar problemas no desenvolvimento. São assim evitados problemas de compatibilidade que possam surgir ao usar

plataformas que não estejam integradas ou que essa mesma integração seja mais difícil e esteja sujeita a erros.

- Popularidade e Documentação - Devido ao período da dissertação não ser muito longo faz com que o tempo disponível para o projeto para desenvolver a aplicação funcional seja relativamente curto, o que faz com que desde o início seja importante encontrar formas de prevenir e evitar os erros ou problemas que podem causar atrasos. É uma excelente ajuda para o desenvolvimento ter uma documentação bem estruturada assim como uma comunidade ativa e dúvidas já resolvidas partilhadas em fóruns.

5.2.1 Apache Cordova

A framework escolhida para fazer o embrulho da aplicação e torna-la legível pelas plataformas móveis é o Apache Cordova [28]. Na figura 5.1 está representada a arquitetura de alto nível sobre o funcionamento desta ferramenta.

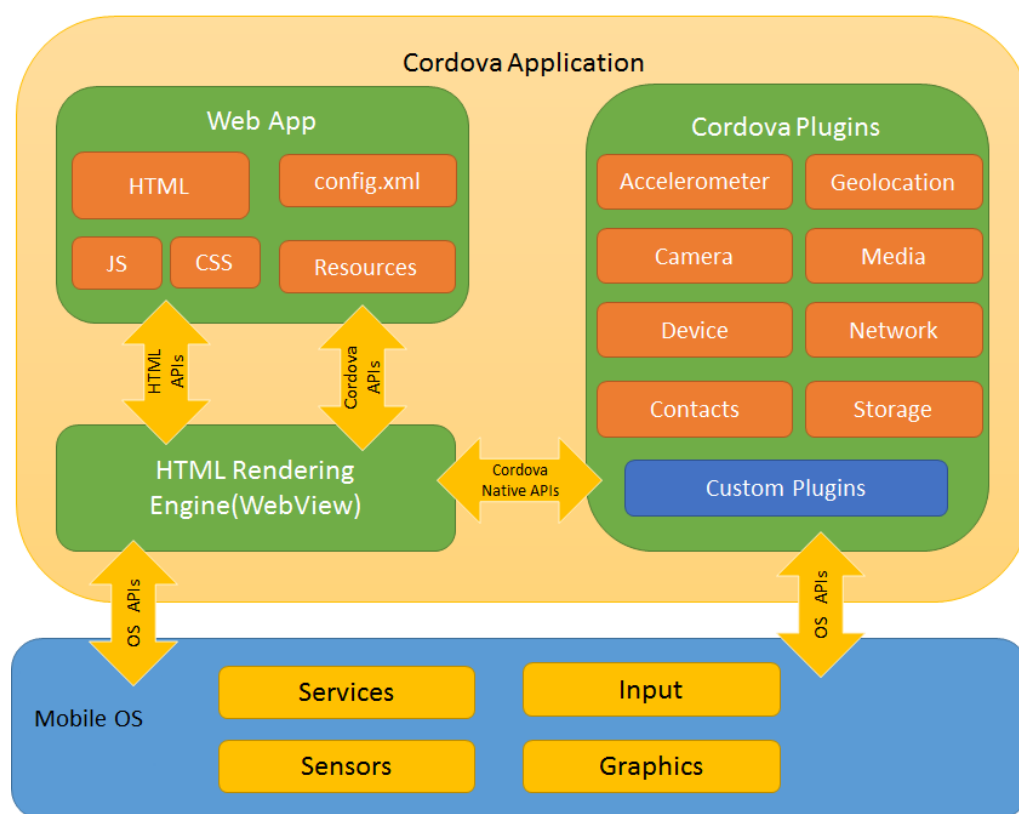


Figura 5.1: Arquitetura do Apache Cordova [3]

Da figura 5.1 é possível identificar as principais áreas de atuação da framework [3].

- **WebView** - A principal funcionalidade da WebView é fornecer à aplicação a sua UI. Em algumas plataformas pode ser uma componente extensa dentro da aplicação híbrida que misturam a WebView com componentes nativos do dispositivo.

- WebApp - É onde reside todo o código da aplicação, a mesma é implementada como uma página Web com o ficheiro *index.html* por defeito que faz referência ao CSS, JavaScript, imagens ou outros recursos necessários para correr a aplicação, que necessita do ficheiro *config.xml* para fornecer as especificações da aplicação.
- Plugins - Parte integrante do Cordova. Possibilitam as comunicações entre a interface com os componentes nativos e torna a aplicação compatível com a API do dispositivo móvel. Assim é possível invocar código nativo com JavaScript.

5.2.2 Ionic

A ferramenta de UI escolhida foi construída com base em AngularJS e em cima da framework Cordova, já explicada. Na sua página da internet [29] são definidas diferentes categorias para quem a vai usar, desde a introdução à ferramenta, a API que contem o código para se utilizar botões, menus e barras de navegação por exemplo, assim como fórum e perguntas mais recorrentes feitas por programadores. A framework fornece diversas ferramentas e serviços para quem desenvolve uma aplicação híbrida, permitindo o uso de vários componentes [30].

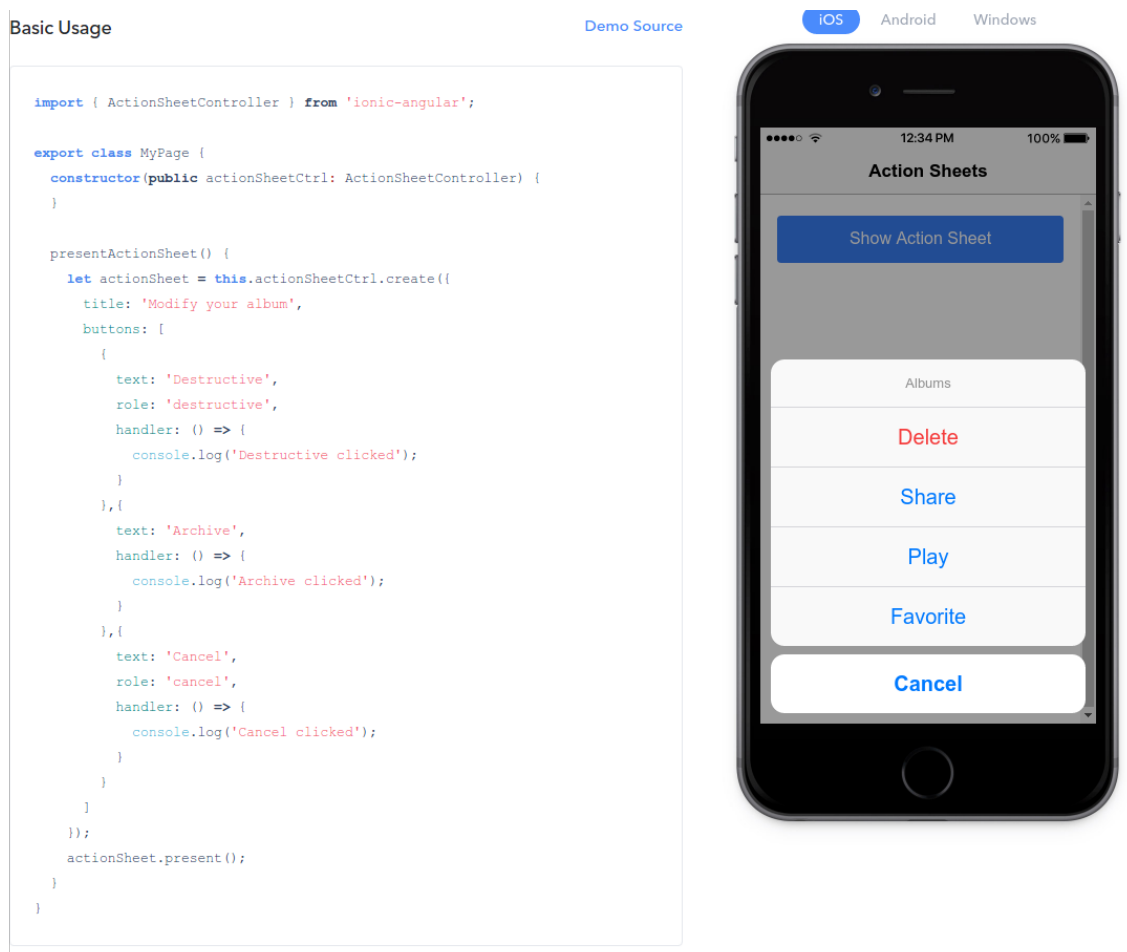


Figura 5.2: Exemplo de utilização da Ionic [4]

5.2.3 Yii

Embora o estudo elaborado em 3.3.3 indicasse as frameworks PHP Laravel e Symfony como melhores escolhas, a IPBrick requereu a utilização da Yii [31] que já está implementada em todos os seus sistemas e todos os colaboradores já estão familiarizados com a mesma o que faz com o tempo de aprendizagem seja curto. Refira-se ainda o facto de ter apenas uma desvantagem facilmente contornável, nomeadamente a complexidade do código que se resolve com recurso a comentários.

5.3 Arquitetura

5.3.1 Alto Nível

Estruturada a aplicação e definidas as frameworks começou-se a implementação recorrendo à linguagem de Engenharia de Software UML. Elaborou-se um diagrama de classes para descrever o modelo da aplicação a ser desenvolvida, representado na figura 5.3. A arquitetura segue a estrutura pensada, onde na parte superior do diagrama temos o backend limitado pela classe *Services*, onde se dá o início do frontend, esta comunica com a classe *App* e termina nas classes que compõe a *Action*. A classe *App* foi uma consequência da utilização da framework Ionic que tem a mesma como raiz do projeto. Este modelo permite transmitir uma visão geral da estrutura e relação entre classes.

Foi elaborado também o modelo de implementação, onde foram ocultados alguns detalhes, com os métodos e variáveis utilizadas de cada classe que por ser extenso foi colocado em anexo A.1.

A grande maioria das aplicações, como é o caso do iPortalDoc Mobile, é complexa e possui uma elevada quantidade de variáveis, métodos e objetos o que torna difícil descrever o seu comportamento sequencial de forma rápida e intuitiva. Assim, foram desenvolvidos, utilizando a mesma linguagem de engenharia de software, alguns diagramas de sequência que mostram a evolução temporal pretendida da aplicação em vários casos de uso.

O diagrama de sequência da figura 5.4 descreve o início da aplicação onde a raiz, o ficheiro *index.html*, instancia o método *constructor* da classe *APP*, que é sempre o primeiro método a ser chamado quando se arranca uma classe de uma página num projeto Ionic. Segue-se a verificação, com recurso ao *userService*, se já existem guardados os dados do utilizador no *Local Storage*, onde ficam armazenadas as cookies da aplicação e, no caso de existirem os dados, é definida então a Língua e a fotografia a mostrar. É utilizado novamente um método da classe *Services* para averiguar se já foi efetuada autenticação na aplicação, senão o utilizador é encaminhado para a página de login, se sim é-lhe permitido trabalhar no iPortalDoc Mobile, com início na *Home*.

Por serem demasiado específicos e longos, os outros diagramas foram colocados em anexos os diagramas referentes ao *Login* em B.1, *Home* em B.4, *List Actions* em B.5, *Info* em B.10, *Schedule* em B.12, *Chronological Scheme* em B.14, *Mark* em B.16 e *Action* em B.18. Foi utilizada com alguma recorrência a nomenclatura de *Reference* que permite dividir diagramas e assim

torná-los mais legíveis. Não foram especificados todos os diagramas devido ao elevado tempo que consomem na sua conceção. Esta decisão não teve nenhum impacto negativo porque foram utilizados nomes elucidativos nos diagramas e que resultam numa implementação simples como por exemplo igualar os dados de retorno dos *Services* às variáveis utilizadas nos templates.

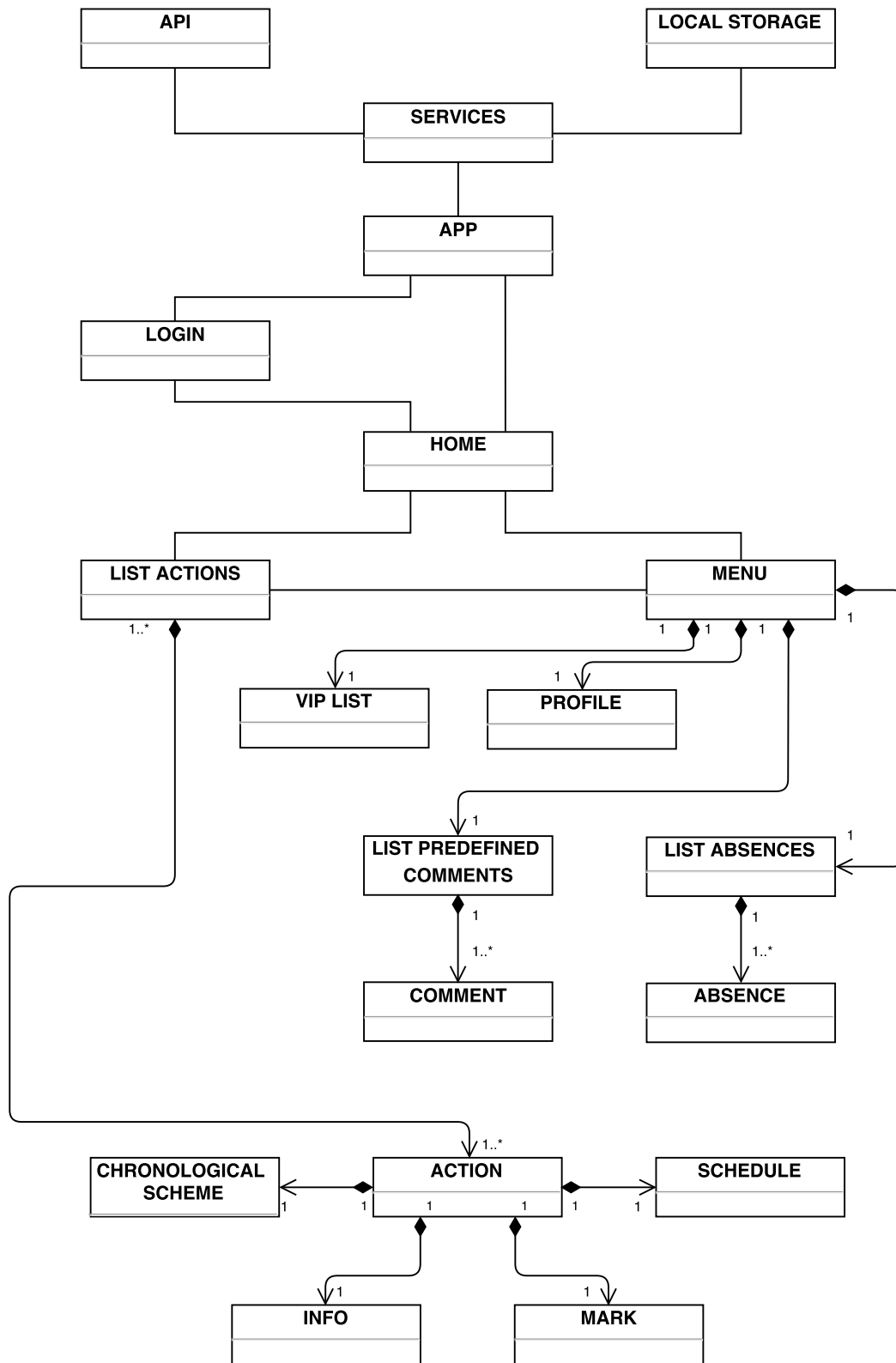


Figura 5.3: Diagrama de Classes - Modelo de especificação

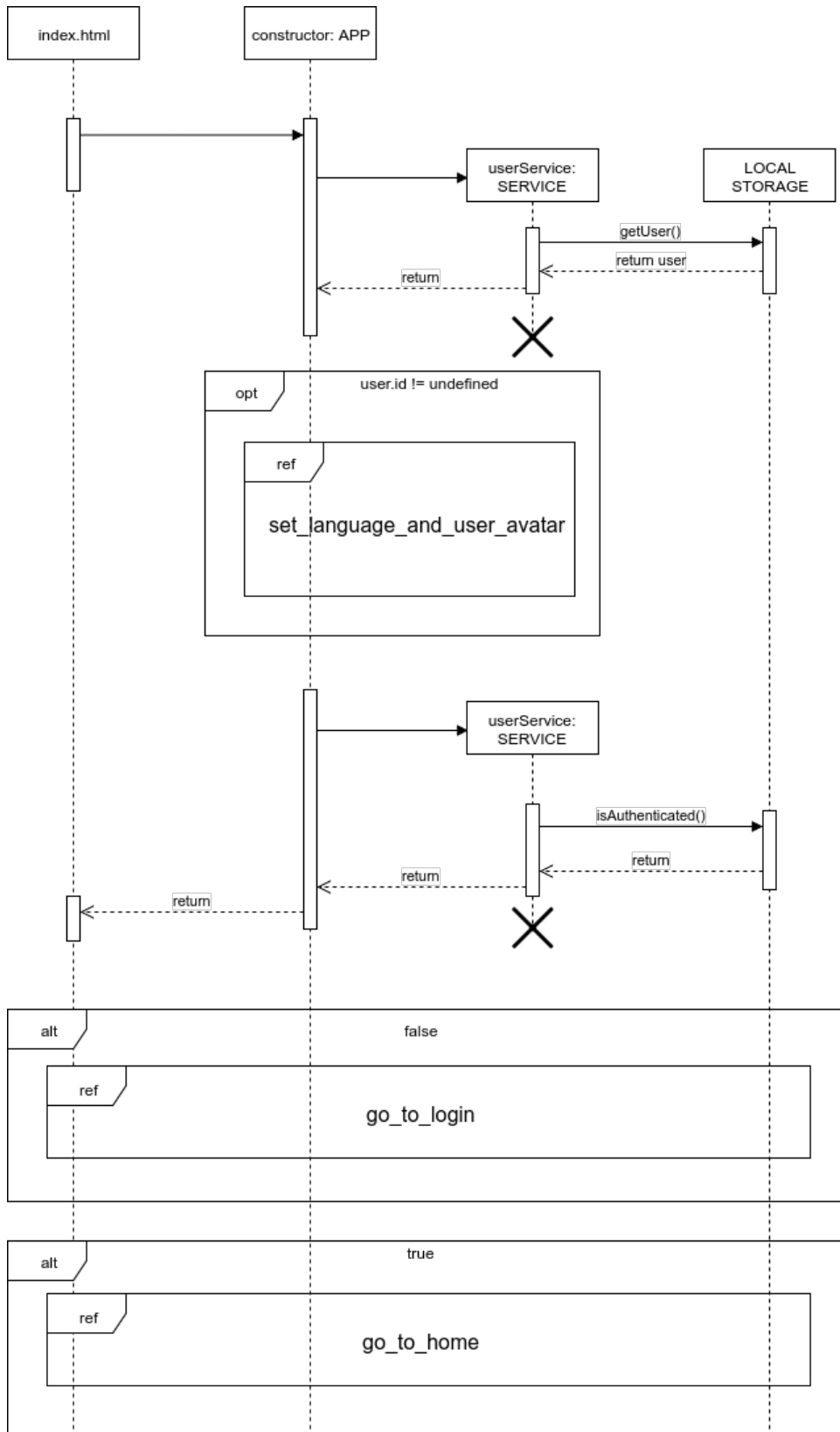


Figura 5.4: Diagrama de Sequência - Início da aplicação

5.3.2 Baixo Nível

Com base na estrutura dos projetos que utilizam a framework Ionic projetou-se uma diretoria API, onde vão estar incluídos os diversos controladores com o objetivo de estabelecer a comunicação com o servidor. Definiu-se que seria necessária uma diretoria de frontend, constituída por uma diretoria de componentes de UI fornecidos pela framework e outra onde estará maior parte do código desenvolvido relativamente as páginas da aplicação. As páginas foram projetadas seguindo parte do princípio MVC (Modelo-Visão-Controlador) [32]. A lógica apenas não é exatamente ao igual princípio porque o iPortalDoc Mobile foi pensado para o frontend estar isolado do backend, portanto não existe o mesmo conceito de modelo. Os modelos da aplicação vão ser implementados na API. A arquitetura segue assim uma lógica multicamada constituída por uma de interface, que corresponde a um ficheiro que define o template (Visão) da página, outro ficheiro que define o estilo da página e por uma camada de controlo (Controlador). Esta é responsável por fazer a ligação entre os vários ficheiros, definir métodos de cada classe e comunicar com os modelos da API. Desta forma, como exemplo, para a diretoria relativa à Lista de Ações a arquitetura definida implica existência de três ficheiros:

- list-actions.html
- list-actions.scss
- list-actions.ts

5.4 Implementação

Definida a estrutura e arquitetura da aplicação, a transição para a implementação não foi complexa porque as variáveis e métodos a utilizar estavam já caracterizados na sua maioria no diagrama de classes e a lógica temporal de funcionamento definida no diagrama de sequência. Como exemplo, algumas páginas da aplicação ficaram estruturadas da seguinte forma:

- action
 - action.html
 - action.scss
 - action.ts
- info
 - info.html
 - info.scss
 - info.ts
- list-actions

- list-.html
- list-actions.scss
- list-actions.ts
- login
 - login.html
 - login.scss
 - login.ts
- user-absence
 - user-absence.html
 - user-absence.scss
 - user-absence.ts
- user-commentlist
 - user-commentlist.html
 - user-commentlist.scss
 - user-commentlist.ts
- user-vip
 - user-vip.html
 - user-vip.scss
 - user-vip.ts

O passo seguinte consistiu em desenvolver o código da aplicação nas suas partes de frontend e backend simultaneamente. Foram seguidas a estrutura e arquitetura projetadas, a lógica de multi-camada, e a criação de serviços e API para fazer com que as páginas interajam com o servidor e o armazenamento local. Nas figuras seguintes são exemplificados os ficheiros de frontend e de backend. Algumas partes do código foram ocultadas para facilitar a demonstração devido à extensão do mesmo.

```

<ion-header>
  <ion-navbar hideBackButton>
    <ion-title class="nav-title">{{ title }}</ion-title>
    <ion-buttons end>
      <button clear menuToggle class="menu-button">
        
      </button>
    </ion-buttons>
  </ion-navbar>
  <!-- CODE IN THIS AREA WAS HIDDEN FOR DEMONSTRATION PURPOSE -->
<ion-content>
  <!-- CODE IN THIS AREA WAS HIDDEN FOR DEMONSTRATION PURPOSE -->
  <ion-list>
    <ion-item-sliding #slide *ngFor="let action of actions; let i = index;" >
      <ion-item text-wrap >
        <ion-label>
          <ion-grid>
            <ion-row (click)="goToAction(actions, i)">
              <ion-col width-10>
                <i *ngIf="!action.action_was_accessed" class="fa fa-circle unread" aria-hidden="true"></i>
              </ion-col>
              <ion-col width-50 text-nowrap class="actionparent_performername">
                <strong>{{ action.actionparent_performername }}
                  <span *ngIf="action.actionparent_isuservip == 1">{{'VIP' | translate}}</span>
                </strong>
              </ion-col>
            </ion-row>
          </ion-grid>
        </ion-label>
      </ion-item>
    </ion-list>
  <!-- CODE IN THIS AREA WAS HIDDEN FOR DEMONSTRATION PURPOSE -->
<ion-content>
<ion-footer>
  <!-- CODE IN THIS AREA WAS HIDDEN FOR DEMONSTRATION PURPOSE -->
  <ion-col width-25 *ngIf="countItemsChecked > 0">
    <button (click)="execute(items); executeActive();">
      <i class="fa fa-wrench fa-2x" ></i><label>{{'Execute' | translate}}</label>
    </button>
  </ion-col>
  <!-- CODE IN THIS AREA WAS HIDDEN FOR DEMONSTRATION PURPOSE -->
</ion-footer>

```

Figura 5.5: Exemplo da implementação de um ficheiro de template HTML

```
page-action {  
  
    /*CODE IN THIS AREA WAS HIDDEN FOR DEMONSTRATION PURPOSE */  
  
    .text-input-ios, .text-input-md {  
        margin: 0! important;  
        padding: 2%;  
        width: 100%;  
        border: 1px solid gray;  
        border-radius: 1em;  
    }  
  
    .date {  
        ion-datetime{  
            padding: 3% 3% 3% 3%;  
            border: 1px solid gray !important;  
            border-radius: 1em;  
            color: black;  
            width: 40%;  
            text-align: right;  
        }  
        ion-label{  
            padding: 3% 3% 3% 3%;  
            border: 1px solid gray !important;  
            border-radius: 1em;  
            background-color: gray;  
            color: white;  
        }  
    }  
  
    /*CODE IN THIS AREA WAS HIDDEN FOR DEMONSTRATION PURPOSE */  
  
}
```

Figura 5.6: Exemplo da implementação de um ficheiro de estilo SCSS

```

import { Component } from '@angular/core';
import { NavController, NavParams } from 'ionic-angular';
import { ChronologicalschemePage } from '../chronologicalscheme/chronologicalscheme';
import { DocumentService } from '../../../services/document-service';
import { Loading } from '../../../controllers/message-controller';
// CODE IN THIS AREA WAS HIDDEN FOR DEMONSTRATION PURPOSE
@Component({
  selector: 'page-info',
  templateUrl: 'info.html',
  providers: [DocumentService, Loading]
})
export class InfoPage {
  title: string; items: any[]; document = {tituloformatted: ''}; public item: any;
  constructor(public navCtrl: NavController, public NavParams: NavParams, public viewController: ViewController,
    public documentService: DocumentService, public loading: Loading) {
    this.item = this.NavParams.get("item");
    let document = this.document;
    if ("undefined" !== typeof this.item.document) { //from action list
document = this.item.document;
this.info({id: this.item.docid});
    }
    public info(params) {
      this.loading.show({});
      this.documentService.info(params).subscribe(
        data => {
          this.items = [];
          for (let a of data.json().list) {
            this.items.push(a);
          }
          this.title = data.json().title;
          this.loading.dismiss();
        }); // CODE IN THIS AREA WAS HIDDEN FOR DEMONSTRATION PURPOSE
    }
    goToChronologicalScheme(item){
      this.navCtrl.push(ChronologicalschemePage, { item: item });
    } // CODE IN THIS AREA WAS HIDDEN FOR DEMONSTRATION PURPOSE
  }
}

```

Figura 5.7: Exemplo da implementação de um ficheiro de controlo TS

```
import { Injectable } from '@angular/core';
import { Observable } from 'rxjs/Observable';
import 'rxjs/Rx';
import { HttpService } from '../services/http-service';
import { ConfigService } from './config-service';

@Injectable()
export class DocumentService {
  constructor (public configService: ConfigService, public http: HttpService) {
  }

  private _http(method: string, params): any {
    return this.http.post(this.configService.getVar('server_url') + '/document/' + method, params);
  }

  public info(params) {
    return this._http('info', params);
  }

  public savemark(params) {
    return this._http('savedocumenttag', params);
  }

  public documenttag(params) {
    return this._http('documenttag', params);
  }

  public removedocumenttag(params) {
    return this._http('removedocumenttag', params);
  }
}
```

Figura 5.8: Exemplo da implementação de um serviço em PHP

```
<?php
namespace api\controllers;
use Yii; use yii\filters\ContentNegotiator; use yii\web\Response; use yii\filters\AccessControl;
use yii\rest\Controller; use yii\filters\auth\HttpBearerAuth; use yii\db\Query;
class DocumentController extends Controller
{
    public function behaviors()
    {
        /*CODE IN THIS AREA WAS HIDDEN FOR DEMONSTRATION PUSPOSE*/
        $behaviors['access'] = [
            'class' => AccessControl::className(),
            'rules' => [
                [
                    'allow' => true,
                    'roles' => ['@'],
                ],
            ],
        ];
        return $behaviors;
    }
    /*CODE IN THIS AREA WAS HIDDEN FOR DEMONSTRATION PUSPOSE*/
    public function actionRemovedocumenttag()
    {
        $params = \Yii::$app->getRequest()->getBodyParams();
        $params["iddoc"] = $params["iddoc"] + 0;
        \Yii::$app->db->createCommand()->delete('documentstags', [
            "iddoc" => $params["iddoc"],
            "iduser" => \Yii::$app->user->identity->id
        ]->execute();
        return [];
    }
}
```

Figura 5.9: Exemplo da implementação de um controlador da API em PHP

Capítulo 6

Resultados do iPortalDoc Mobile

Este capítulo descreve o refinamento feito à solução no decorrer do projeto devido a novos requisitos, dá a conhecer os resultados finais de implementação e UX obtidos com a aplicação iPortalDoc Mobile e ainda mostra a análise feita ao cumprimento de requisitos.

6.1 Refinamento da solução

Ao longo do projeto percebeu-se que os requisitos não estavam completamente fechados devido ao feedback obtido em reuniões dos responsáveis da IPBrick com o cliente da aplicação. Das reuniões surgiram pedidos para novas funcionalidades relativas aos documentos dentro da ação. No caso do iPortalDoc a relação dos documentos para as ações é de 1:N, ou seja um documento pode estar associado a N ações mas uma ação só tem implicação sobre um documento. Quando se acede à informação da ação está-se na verdade a aceder à informação do documento e o pedido do cliente ia nesse sentido, disponibilizar uma página para pesquisa e consulta de documentos.

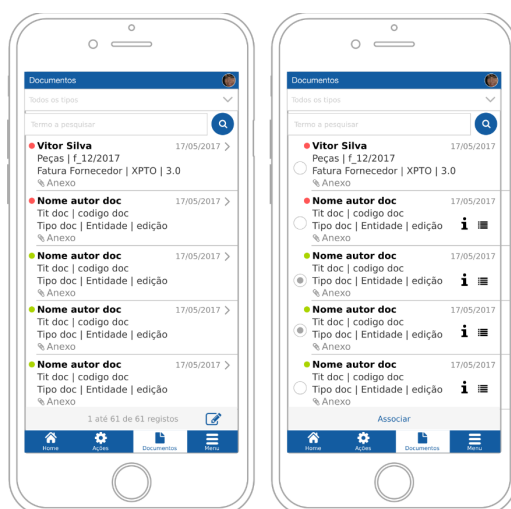
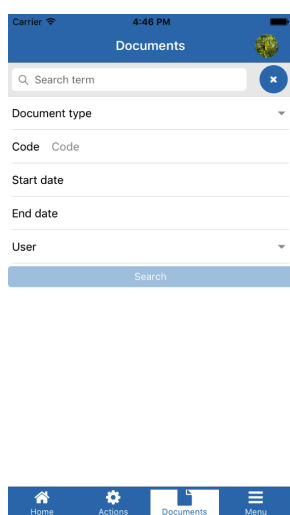
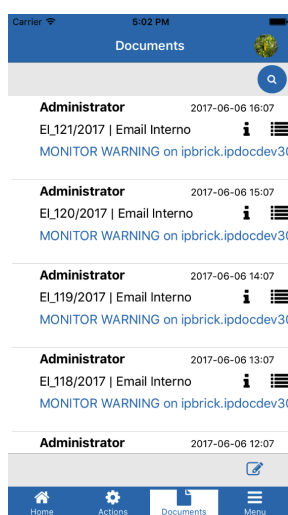


Figura 6.1: Modelo da página de documentos

Foi necessário mudar o *footer* das quatro páginas principais da aplicação: o botão *Edit* foi realocado para dar lugar ao botão *Documents* que permite navegar para uma nova página com o mesmo nome, onde existe uma secção para pesquisa avançada. Depois de preencher os campos que desejar e efetuar a pesquisa, é apresentada ao utilizador uma informação resumida, da mesma forma que na listagem de ações, do ou dos documentos resultantes que podem ser ficheiros de texto e imagens como *.docx* ou *.pdf* ou simplesmente imagens, por exemplo *.png* ou *.jpeg*. Estes avanços obrigaram à utilização de novos componentes da framework ionic, como é o caso das propriedades nativas de download e upload para permitir à aplicação comunicar com o servidor de forma a obter o documento pretendido e assim aceder à *filesystem* do dispositivo para o poder guardar. Para a utilização de propriedades nativas foi ainda necessário passar da implementação e testes no browser para um emulador e depois a um dispositivo real, que implicou algum tempo despendido pela equipa de suporte da IPBrick a instalar novos softwares e a adquirir hardware.



(a) Pesquisa de documentos



(b) Resultados da pesquisa de documentos

Figura 6.2: Resultados da página de Documentos

6.2 Resultados de Implementação

A solução desenvolvida deu origem ao iPortalDoc Mobile testado no browser do computador e vários smartphones, no emulador e por fim num dispositivo real. No entanto não foi possível durante o período de dissertação publicar a aplicação nas lojas para efetuar a distribuição porque para isso é obrigatório ter uma conta de programador na Apple e Android, conta essa que não existia e não era conveniente criar neste momento.

A implementação do iPortalDoc Mobile teve como resultados visíveis as interfaces de utilização. A lógica de multicamada das páginas associada ao funcionamento de backend permitiram atingir resultados concretos que são passíveis de ser exibidos em imagens de sucessivas capturas de ecrã.

A implementação resultou nas páginas:

- Login - Foi possível cumprir os requisitos para guardar dados e dar a possibilidade de personalizar imagem, dentro da página de perfil. Não foi possível garantir o tempo máximo fixado para login devido a terem existidos poucos testes em dispositivos físicos e fora da IPBrick, longe dos servidores.

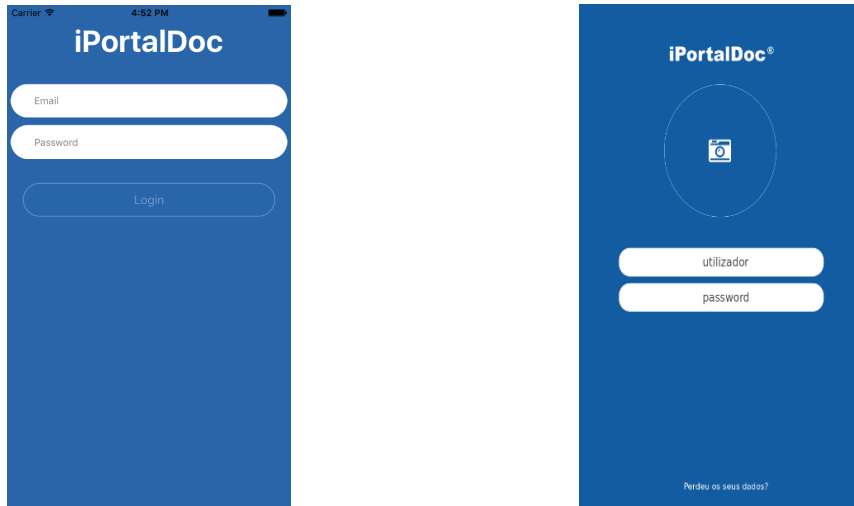


Figura 6.3: Comparação entre o resultado e o modelo da página de Login

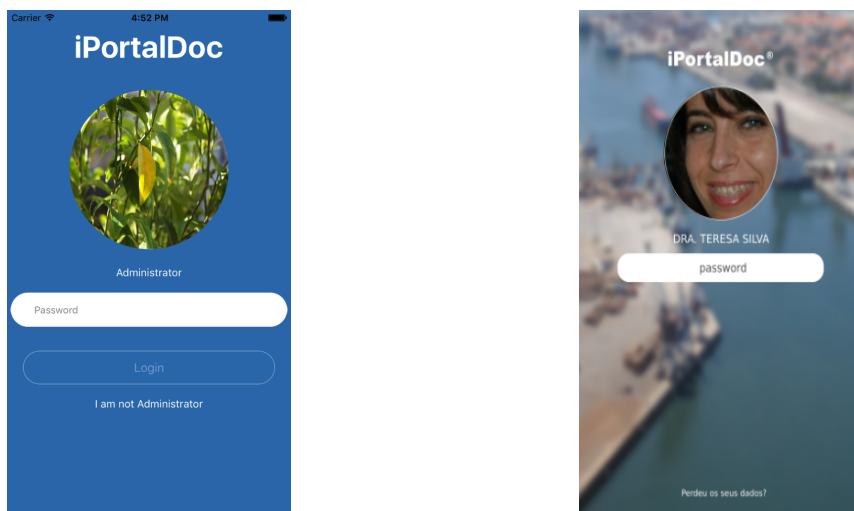


Figura 6.4: Comparação entre o resultado e o modelo da página de Login com informação

- Home

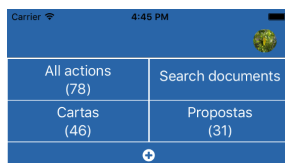


Figura 6.5: Resultado da implementação da página Home

- Menu - Foram cumpridos os requisitos do botão do canto superior direito, personalizar lista VIP, comentários e ausências. Não foi disponibilizada a possibilidade de personalizar os dados pessoais pois no iPortalDoc não está prevista essa funcionalidade de momento.

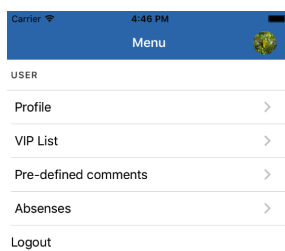


Figura 6.6: Resultado da implementação da página Menu

- Lista de Ações - Cumpriu-se todos os requisitos registrados.

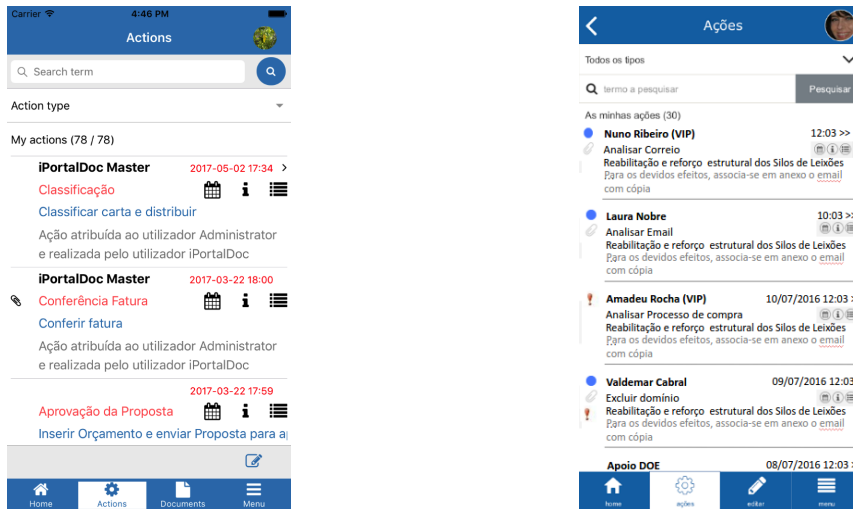


Figura 6.7: Comparação entre o resultado e o modelo da página de Lista de Ações

- Ação - Foram cumpridos todos os requisitos pedidos pelo cliente.

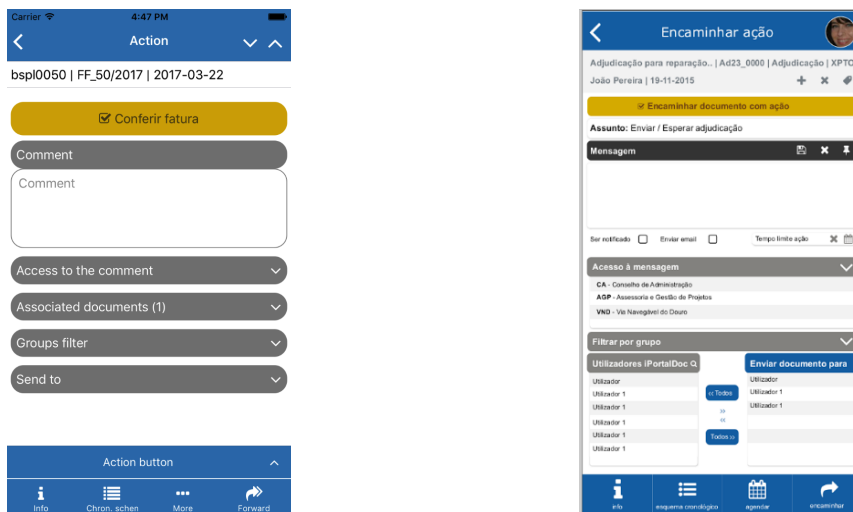


Figura 6.8: Comparação entre o resultado e o modelo da página de Ação

- Info - Todos os requisitos pretendidos foram disponibilizados.

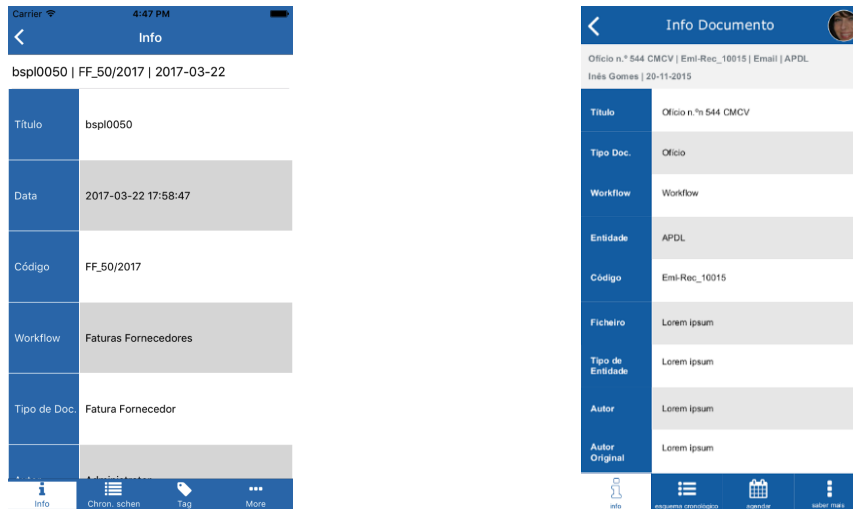


Figura 6.9: Comparação entre o resultado e o modelo da página de Info

- Esquema Cronológico - Esta página cumpre todos os requisitos especificados.

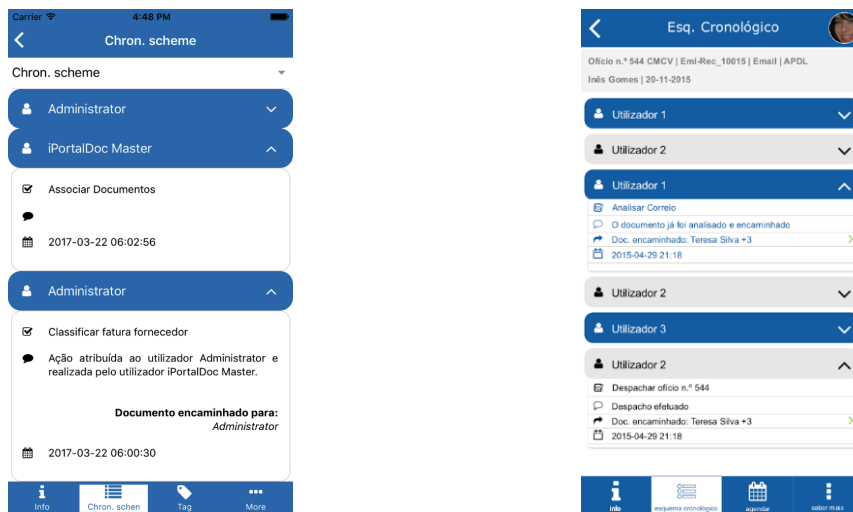


Figura 6.10: Comparação entre o resultado e o modelo da página de Esquema Cronológico

- Agendamento

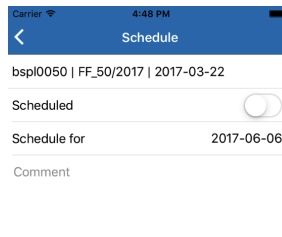


Figura 6.11: Resultado da implementação da página Agendamento

- Marcação

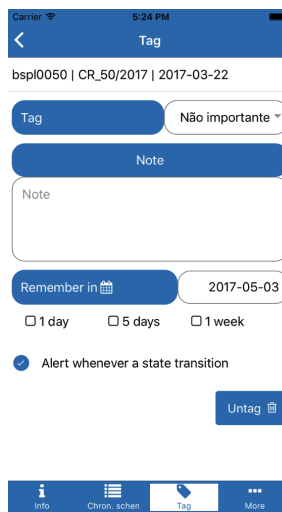


Figura 6.12: Resultado da implementação da página Marcação

- Lista VIP

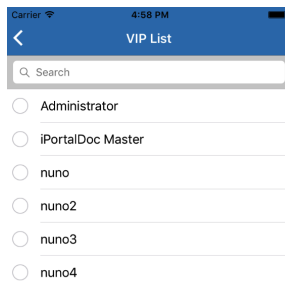


Figura 6.13: Resultado da implementação da página Lista VIP

- Lista de Comentários Pré-definidos

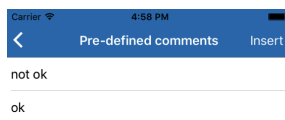


Figura 6.14: Resultado da implementação da página Lista de Comentários Pré-definidos

- Comentário

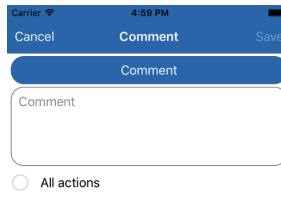


Figura 6.15: Resultado da implementação da página Comentário

- Lista de Ausências

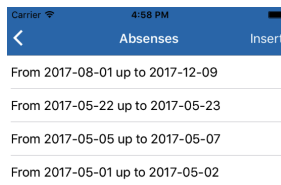


Figura 6.16: Resultado da implementação da página Lista de Ausências

- Ausência



Figura 6.17: Resultado da implementação da página Ausência

6.3 Resultados de User Experience

Outros resultados obtidos são consequência da aplicação de conceitos de UX, nomeadamente, saídas claras, com os botões de regresso e de cancelamento representado na figura 6.18, lógica consistente, no design e cores constantes, gamification, nas cores das mensagens de sucesso e erro representados na figura 6.20, mentalidade, no tamanho do ecrã, dos ícones e controlo, nas figuras 6.22 e 6.23. Nas imagens seguintes obtidas de capturas de ecrã estão alguns exemplos de resultados de UX:

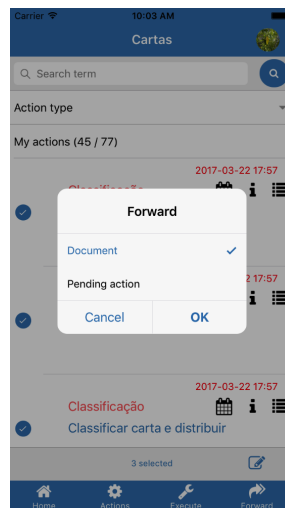


Figura 6.18: Resultado da implementação do conceito de UX: Saídas claras

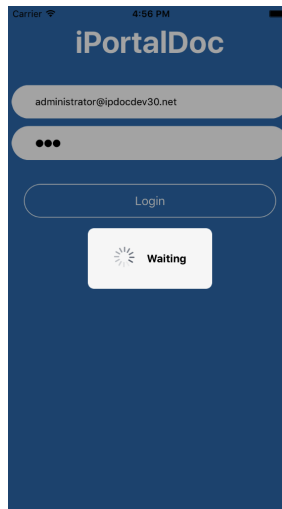


Figura 6.19: Resultado da implementação do conceito de UX: Feedback

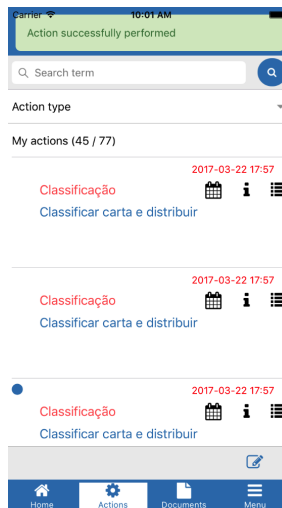


Figura 6.20: Resultado da implementação do conceito de UX: Respostas

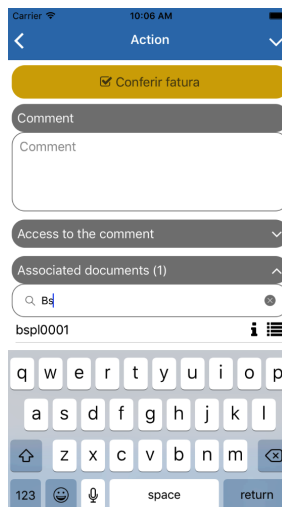


Figura 6.21: Resultado da funcionalidade Pesquisas implementadas

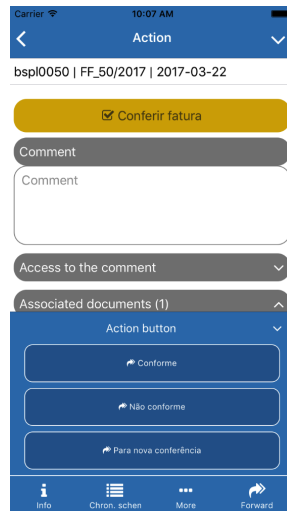


Figura 6.22: Resultado da funcionalidade Localização dos botões de execução implementada

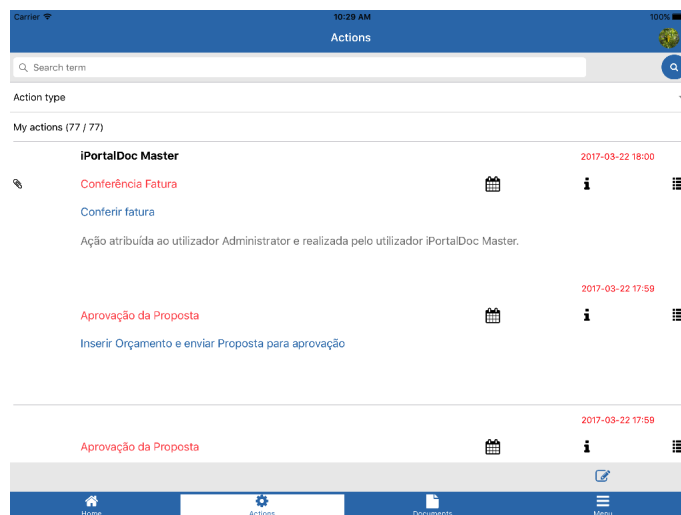


Figura 6.23: Resultado da implementação para uma resolução de tablet (iPad Air)

6.4 Análise do cumprimento requisitos

Uma das formas utilizadas para avaliar o iPortalDoc Mobile é determinar se os seus requisitos foram cumpridos. Os requisitos da aplicação foram divididos em requisitos funcionais, definidos como os objetivos essenciais e se traduzem no modelo multicamada definido na arquitetura 5.3 como o template, o estilo e lógica das páginas, e requisitos não-funcionais que dizem respeito a objetivos específicos que não impedem o funcionamento da aplicação como por exemplo deslizar uma ação no sentido esquerda direita para a marcar como não lida. O cumprimento dos requisitos funcionais foi analisado no decorrer do projeto para perceber a evolução e se estava a ser feita uma boa gestão do esforço e tempo. Como descrito em 5.1.1 e em 6.1, foram definidas 15 páginas de interface com o utilizador que correspondem assim a 45 requisitos funcionais. Acrescentaram-se ainda três requisitos, dois para o teste no browser, como se a aplicação fosse uma página Web, num computador e num dispositivo móvel e outro para o teste efetivo da plataforma criada como aplicação do dispositivo. Do cumprimento de requisitos por data (bi-semanal) foi construído o gráfico final onde se percebe que a implementação do projeto decorreu com normalidade, não necessitando de ações corretoras, como mostra a figura 6.24.

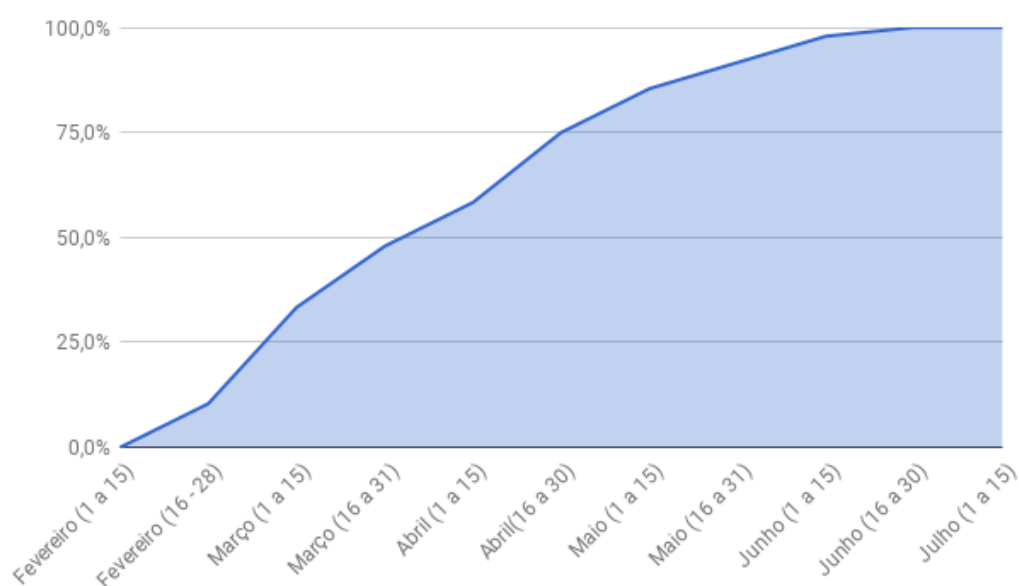


Figura 6.24: Evolução do cumprimento de requisitos funcionais

6.5 Experiências e análise crítica global ao iPortalDoc Mobile

O último dos requisitos a ser cumprido foi o teste do iPortalDoc Mobile como aplicação de um dispositivo real num ambiente controlado. Como já tinha sido constatado no simulador, a framework Apache Cordova possibilitou que a aplicação desenvolvida com recurso a linguagens de programação Web se tornasse legível por um dispositivo móvel de comunicações. Este teste foi efetuado a poucos dias do final da dissertação devido à disponibilização tardia do hardware. Não houve tempo para disponibilizar a aplicação a potenciais consumidores e efetuar uma análise quantitativa de inquéritos de satisfação, no entanto é possível fazer uma análise crítica pessoal à aplicação nas três matérias mais relevantes.

- **Layout** - A integração entre as frameworks Apache Cordova e Ionic e a consequente implementação permitiram obter o desenho inicial do layout no dispositivo sem diferenças evidentes. Todas as expectativas criadas relativas ao design do iPortalDoc Mobile foram alcançadas. Porém constatou-se que a implementação do requisito da página Lista de Ações relativamente à mudança do footer no momento da seleção de ações torna-se pouco intuitivo pois vai contra o princípio de convenções familiares de User Interface. Pode ser estranho para o utilizador ter botões de execução diferentes no footer dependendo do momento e da interação que fez com o sistema.
- **Funcionalidades** - A aplicação atinge o seu propósito pois desempenha com sucesso as tarefas a que se propõe. Notou-se no entanto alguma demora no desempenho e execução de ações. Esta demora pode ser explicada pelo fato do teste ter sido realizado num ambiente controlado com recurso a WiFi e pela existência de uma proxy na comunicação entre o dispositivo e o servidor, ou pela arquitetura implementada. Desta forma, os tempos de resposta obtidos do iPortalDoc Mobile não se revelaram totalmente satisfatórios.
- **User Experience e User Interface** - Nesta matéria salienta-se a importância do conceito aplicado de feedback. Aliado ao fato das execuções serem um pouco lentas, é fundamental mostrar ao utilizador que existem operações em curso senão o resultado mais provável é a insistência na interação efetuada com o sistema. Este resultado pode assim implicar um aumento no tempo de resposta da aplicação. Por outro lado, mensagens de feedback, como a apresentada na figura 6.19, podem tornar-se demasiado intrusivas ao impedir a utilização da interface e provocar algum incómodo ao utilizador quando se tornam recorrentes. Pode ser interessante eliminar estas mensagens quando não são estritamente necessárias e optar em vez disso por colocar um ícone animado [33] no local onde o utilizador interagiu com a aplicação.

Capítulo 7

Conclusão e trabalhos futuros

O projeto abordado por esta dissertação foi totalmente concretizado, com todos os objetivos propostos inicialmente cumpridos.

A nível pessoal, tornou-se numa experiência extremamente agradável e permitiu desenvolver capacidades relativas a princípios e técnicas de engenharia no projeto e implementação de um sistema de software complexo.

O problema de desenvolver uma aplicação mobile com recurso a linguagens de programação Web foi solucionado através da utilização das frameworks escolhidas, Ionic e Apache Cordova.

Todas as páginas da aplicação foram implementadas e ficaram em conformidade com os modelos elaborados. Estas providenciam ao utilizador todas as funcionalidades requisitadas pelo cliente e baseiam-se nos conceitos de User Experience e User Interface.

O desenho da estrutura de frontend e backend aliado ao esboço da arquitetura de alto e baixo nível permitiram transitar rapidamente para a implementação do iPortalDoc Mobile. Em suma, o projeto da aplicação mostrou-se fundamental pois permitiu desenvolver uma aplicação móvel complexa em relativamente pouco tempo.

7.1 Trabalho Futuro

O trabalho relativo ao iPortalDoc Mobile não está terminado porque foram efetuados poucos testes em ambiente real portanto é indispensável, em trabalho futuro, proceder a novas avaliações da aplicação. Estas deverão ser relativas ao design, funcionalidades e tempos de resposta em vários dispositivos reais de comunicações móveis e, caso sejam negativas, é necessário proceder a um estudo e à consequente aplicação de melhorias. Algumas destas foram já detetadas na análise crítica à aplicação. A implementação do requisito especificado pelo cliente sobre a mudança do footer, o desempenho por vezes lento e as mensagens intrusivas de feedback ao utilizador devem ser tidas em conta para futuras melhorias da aplicação no caso do cliente não ser relutante a estas mudanças propostas.

O projeto deverá seguir depois para uma fase exposição e análise do feedback do cliente. Consequentemente o trabalho passará pela inserção de novas funcionalidades advindas de requisitos de clientes ou projetadas pela IPBrick para permitir que no melhor dos casos os utilizadores não tenham a necessidade de recorrer ao seu computador para utilizar alguma funcionalidade do iPortalDoc.

Anexo A

Diagrama de Classes

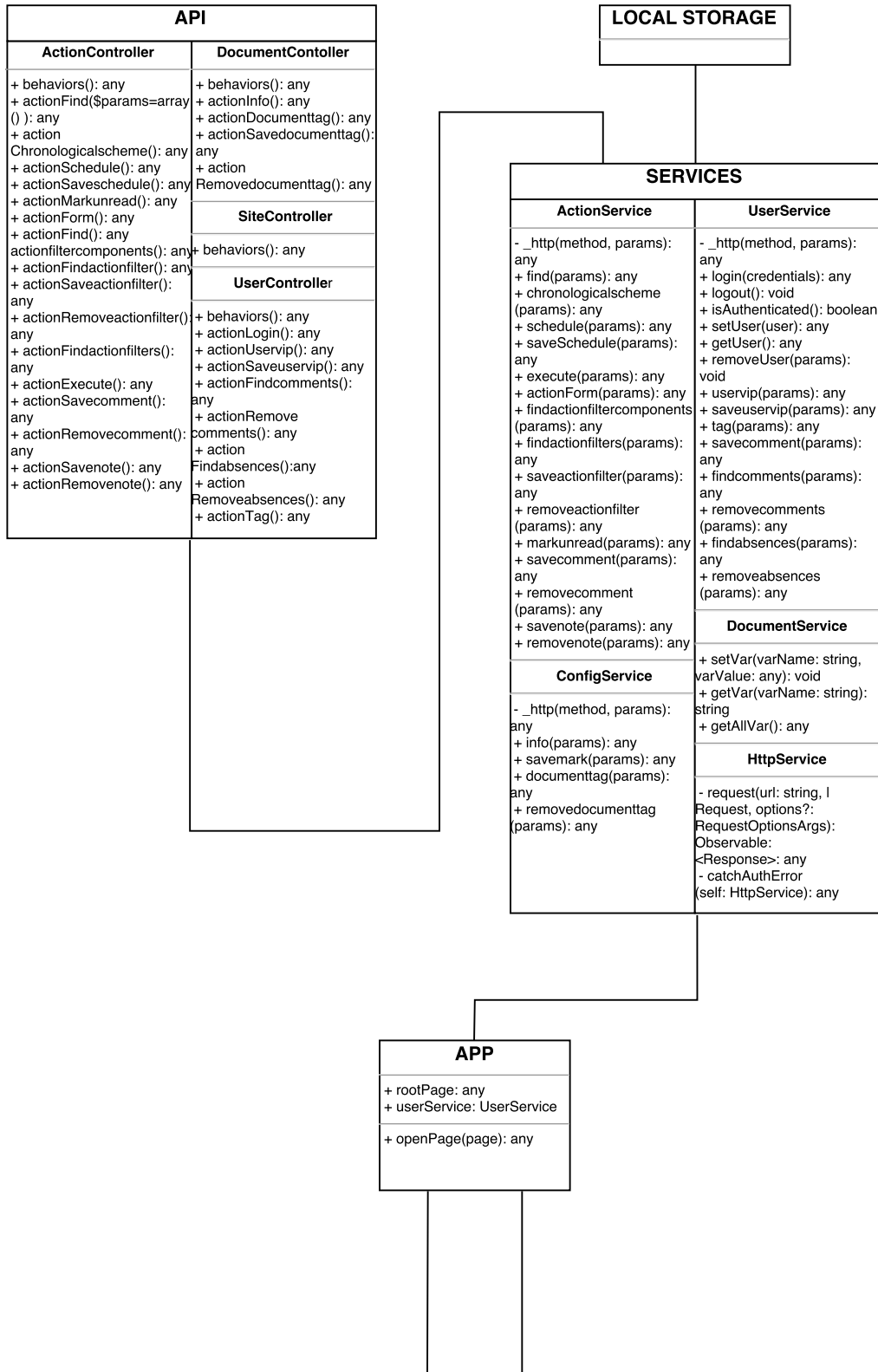


Figura A.1: Diagrama de Classes (1 de 3)

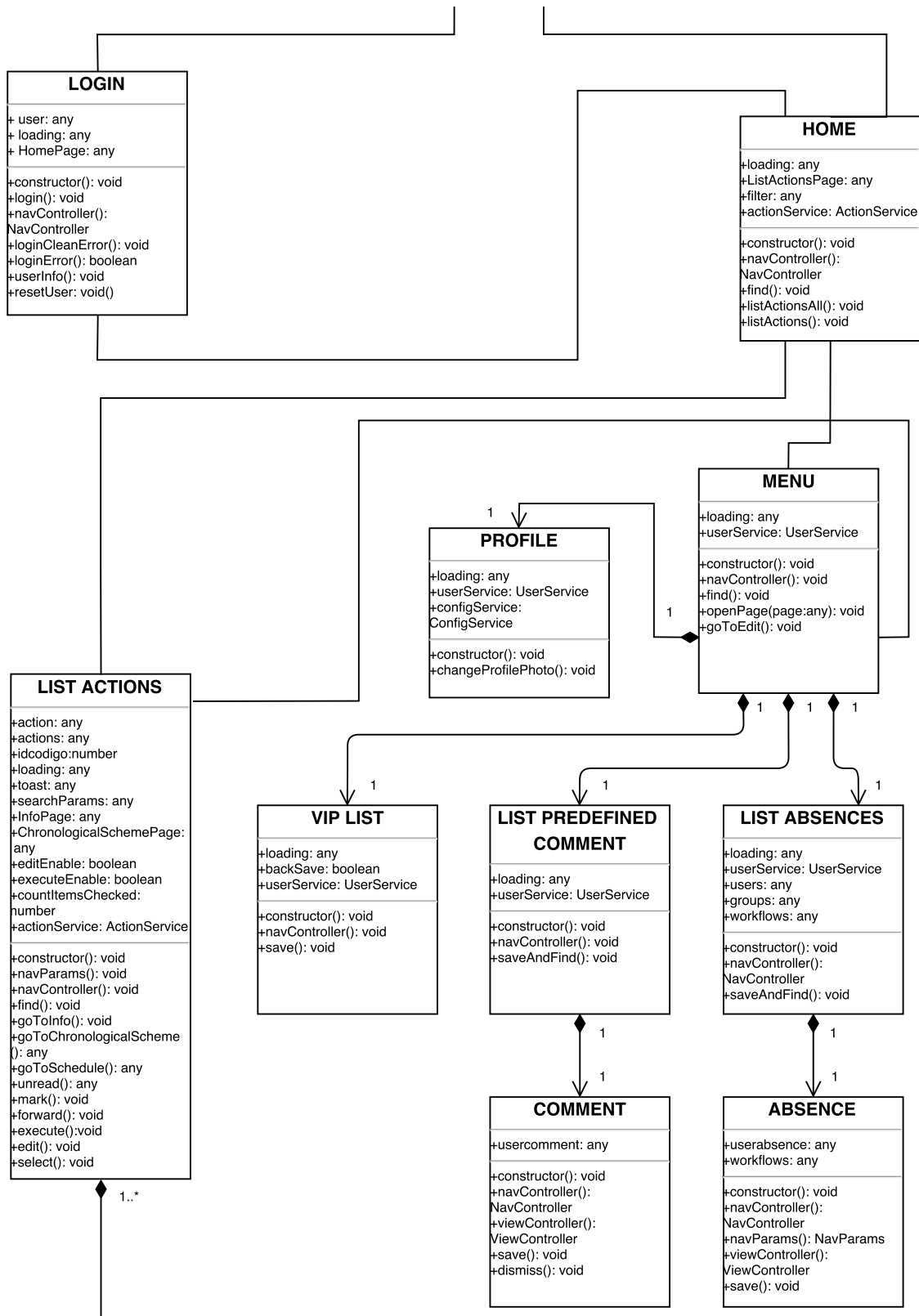


Figura A.2: Diagrama de Classes (2 de 3)

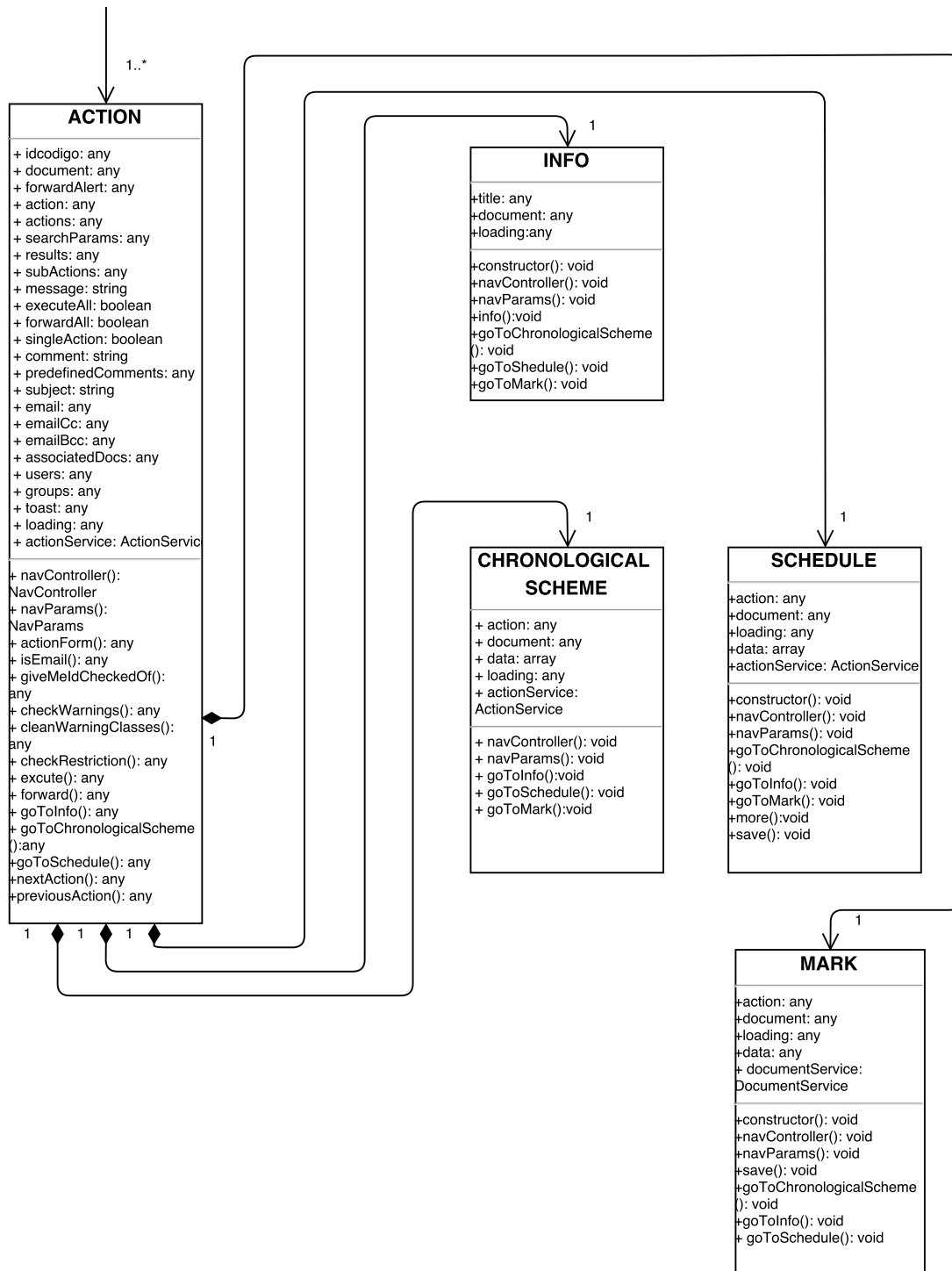


Figura A.3: Diagrama de Classes (3 de 3)

Anexo B

Diagramas de Sequência

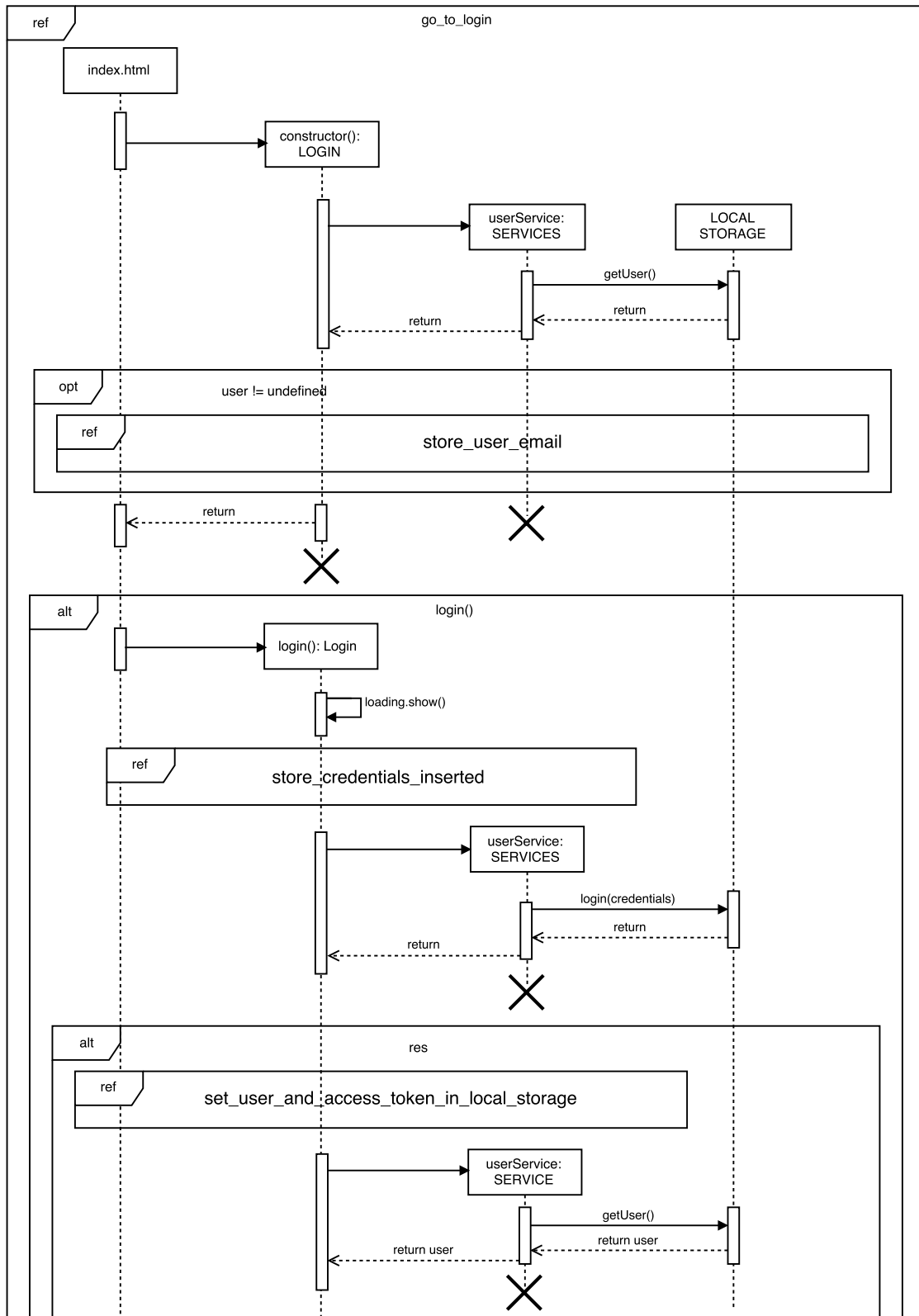


Figura B.1: Diagrama de Sequência - Login (1 de 3)

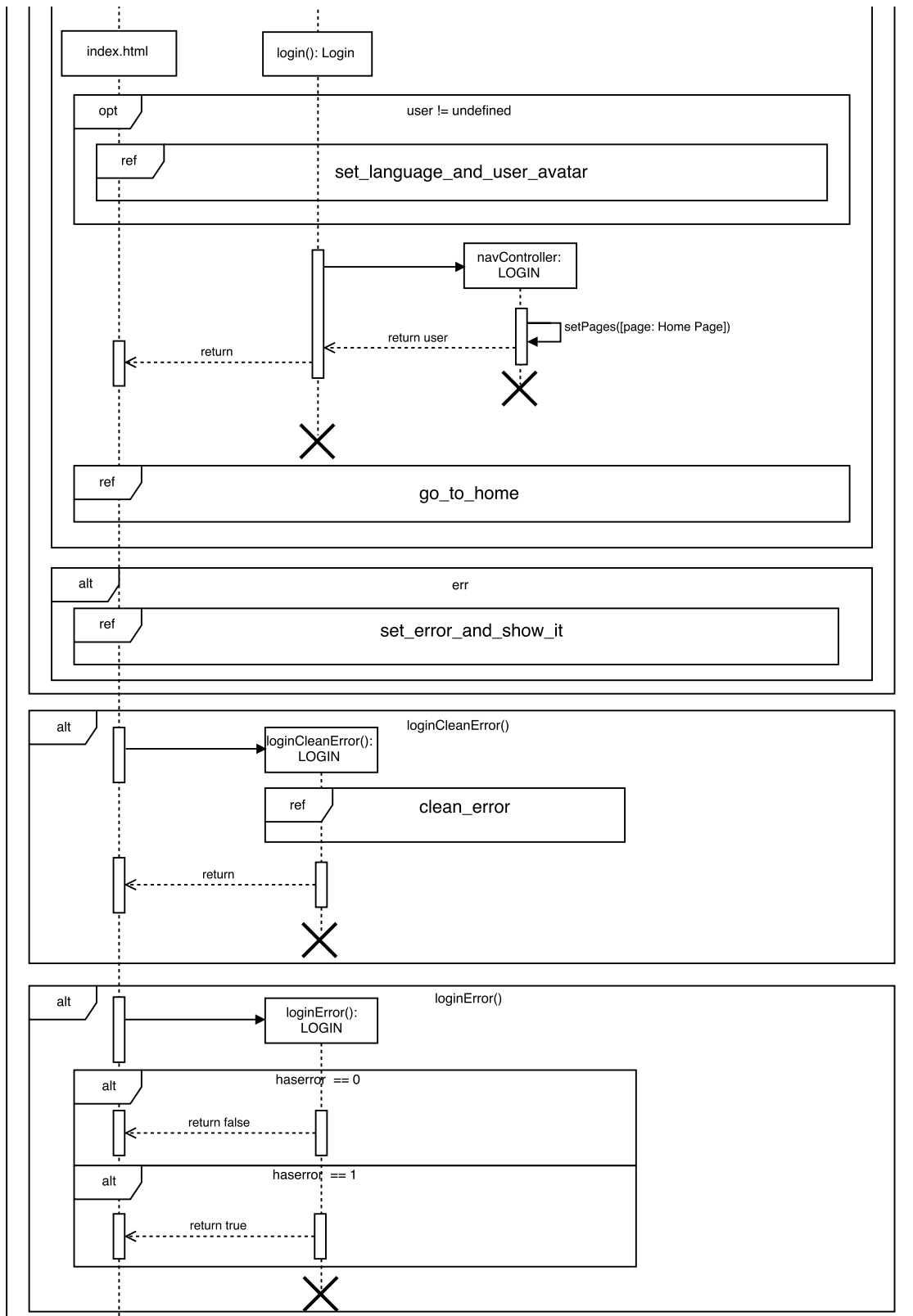


Figura B.2: Diagrama de Sequência - Login (2 de 3)

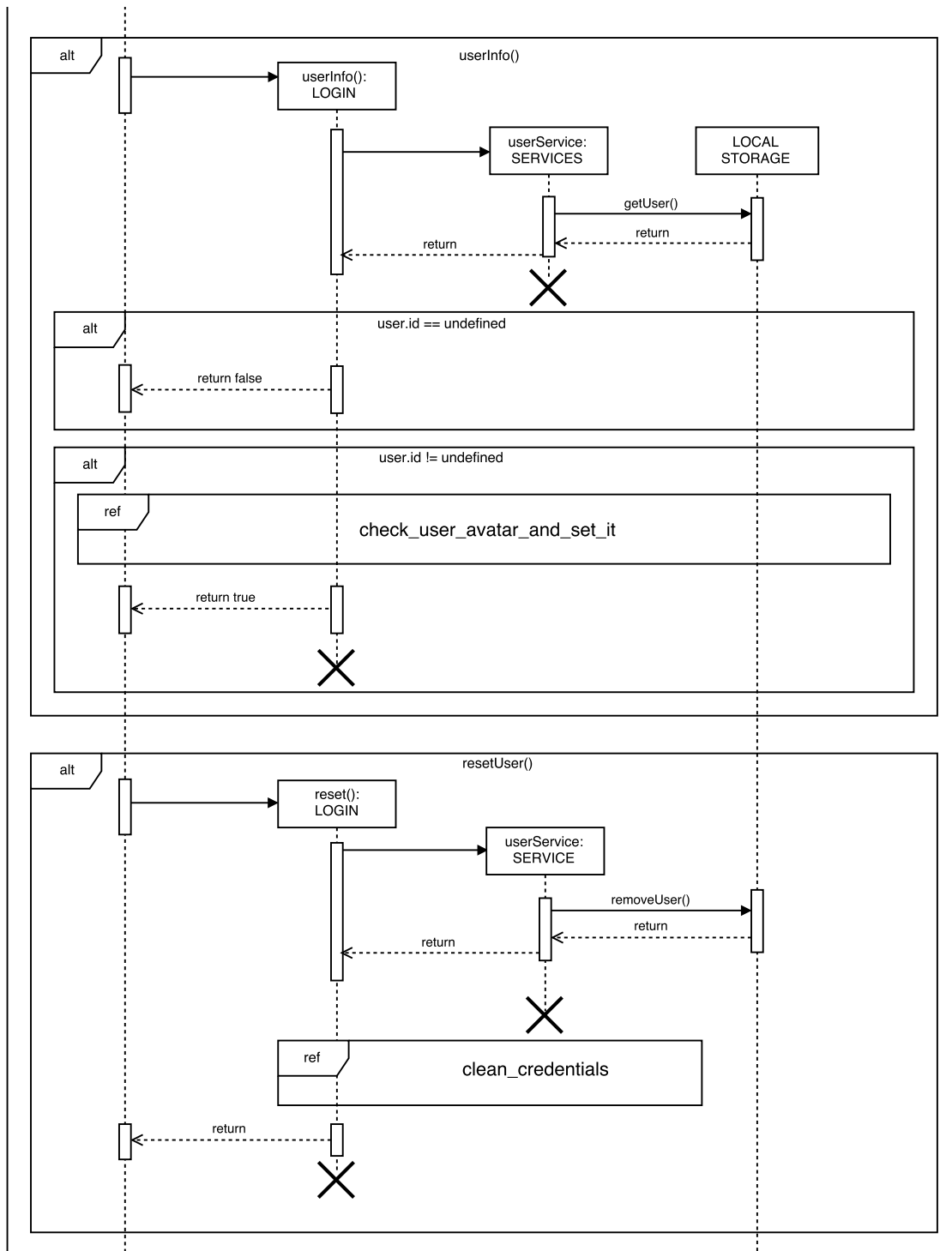


Figura B.3: Diagrama de Sequência - Login (3 de 3)

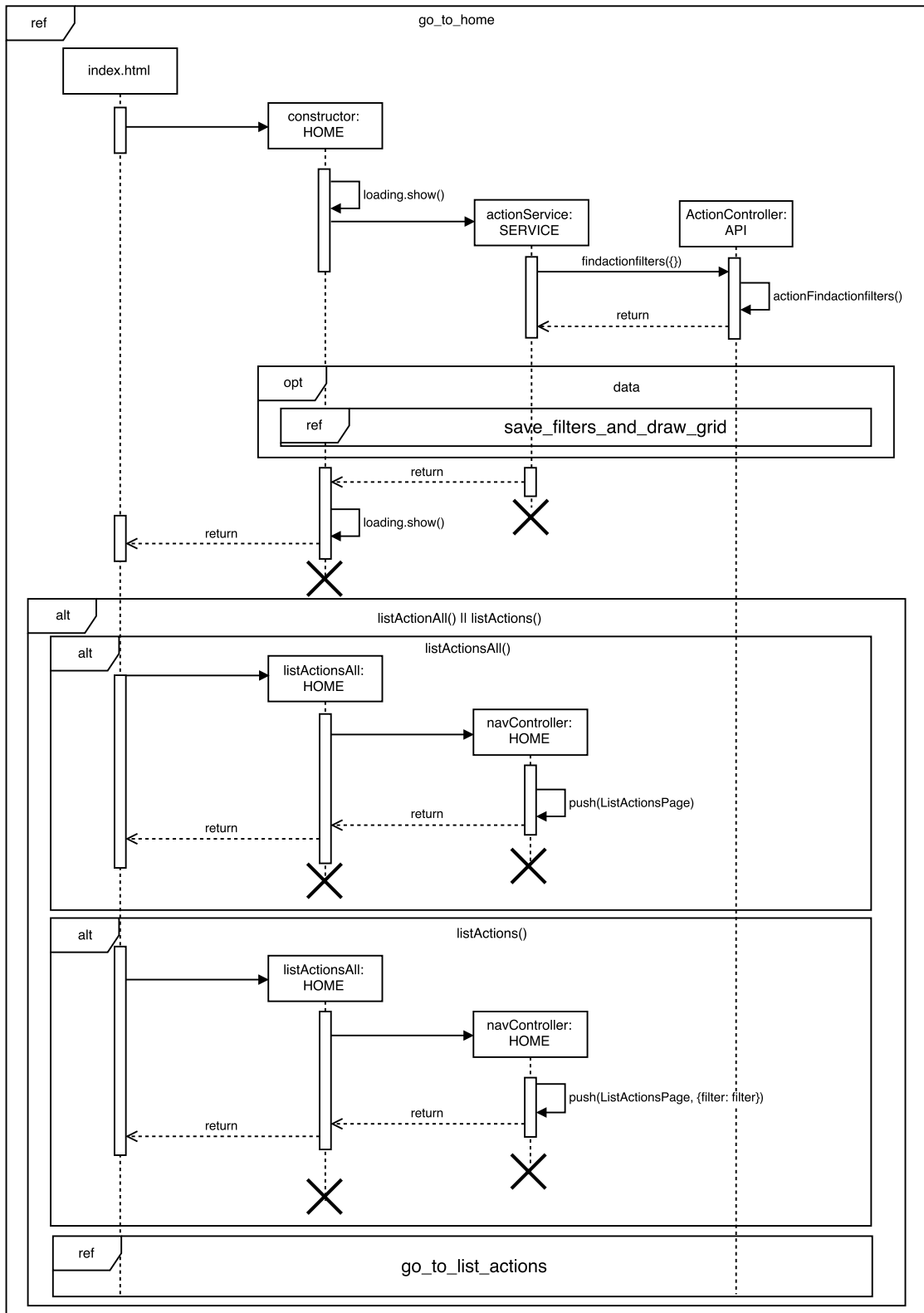


Figura B.4: Diagrama de Sequência - Home

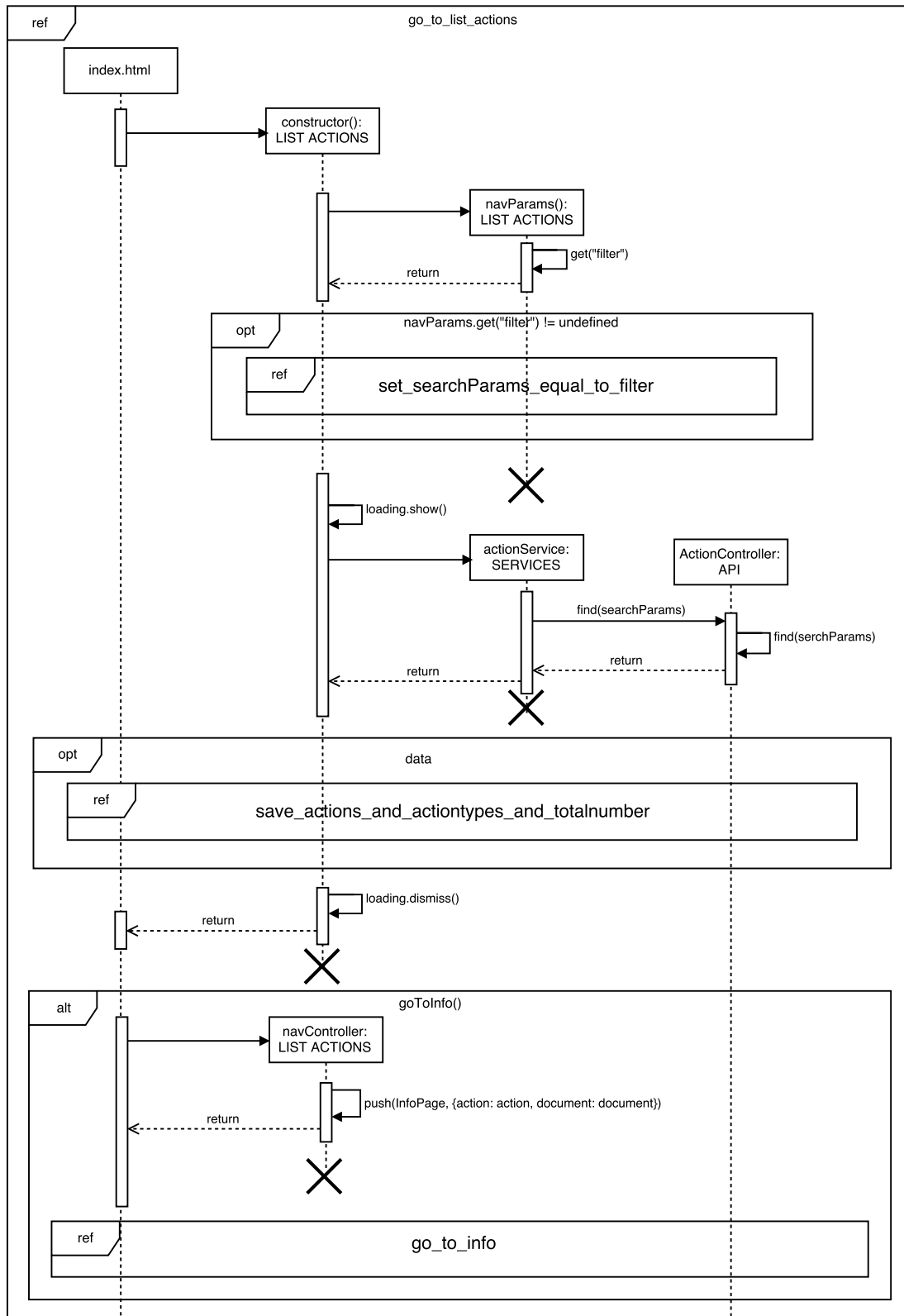


Figura B.5: Diagrama de Sequência - Lista de Ações (1 de 5)

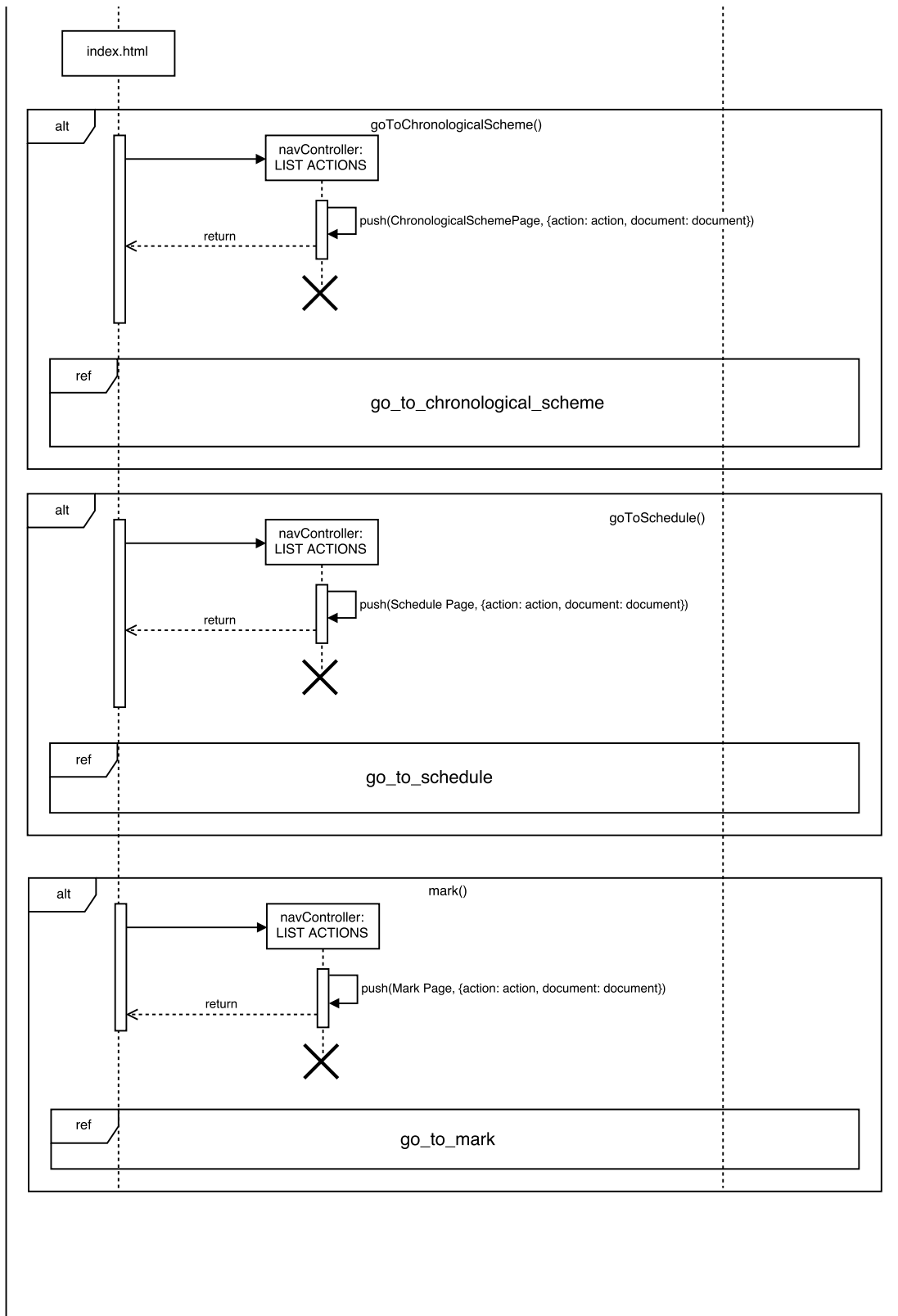


Figura B.6: Diagrama de Sequência - Lista de Ações (2 de 5)

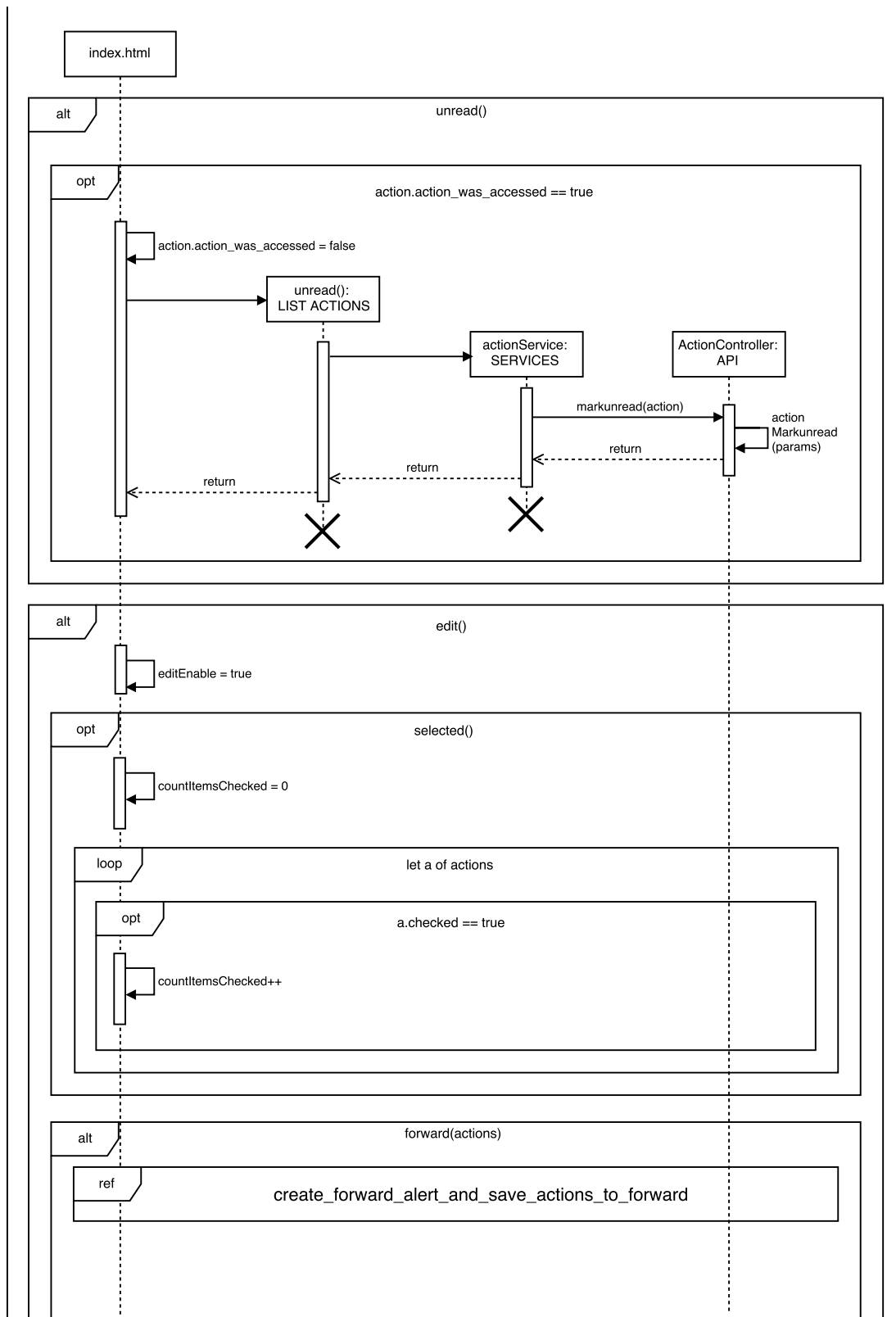


Figura B.7: Diagrama de Sequência - Lista de Ações (3 de 5)

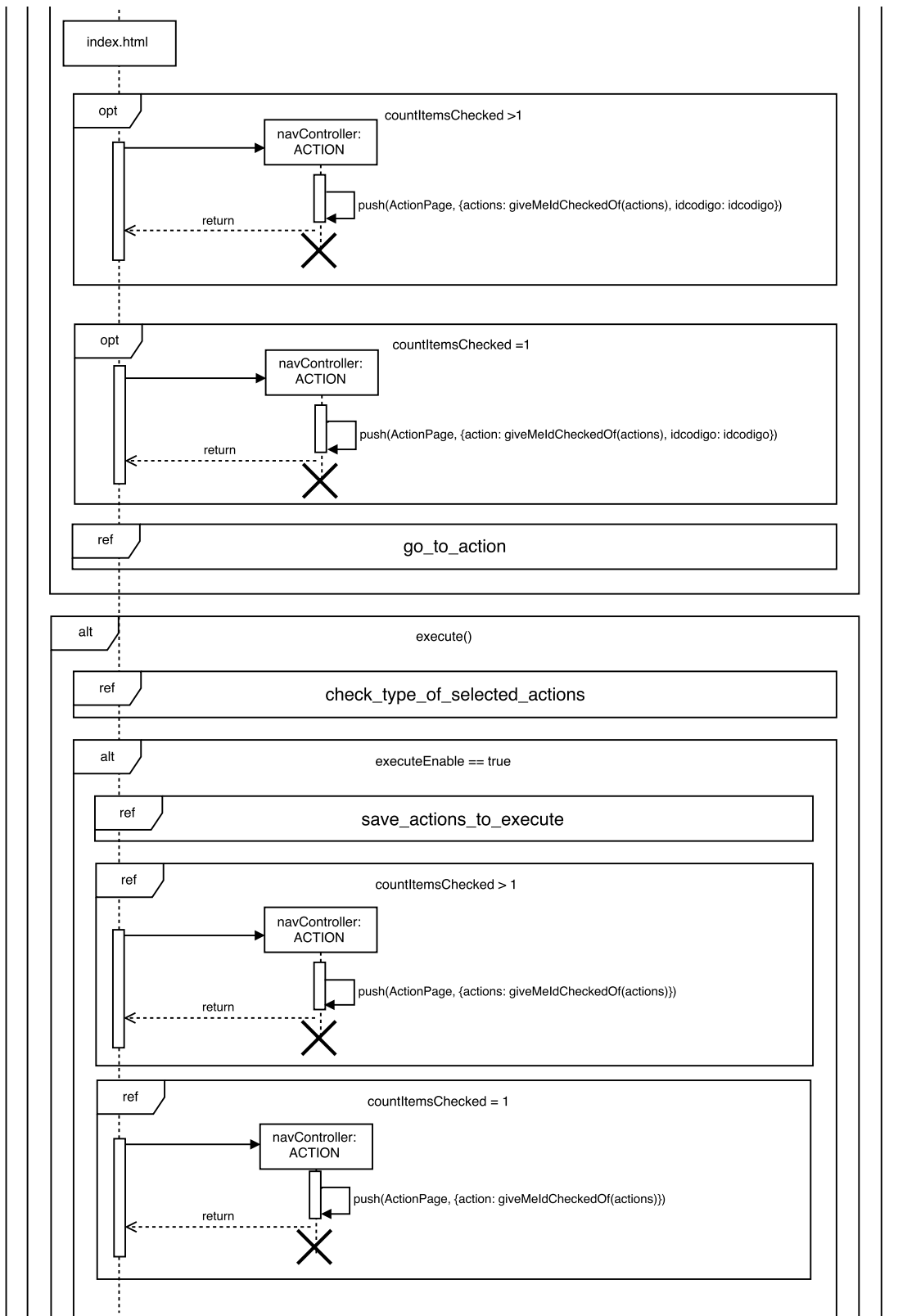


Figura B.8: Diagrama de Sequência - Lista de Ações (4 de 5)

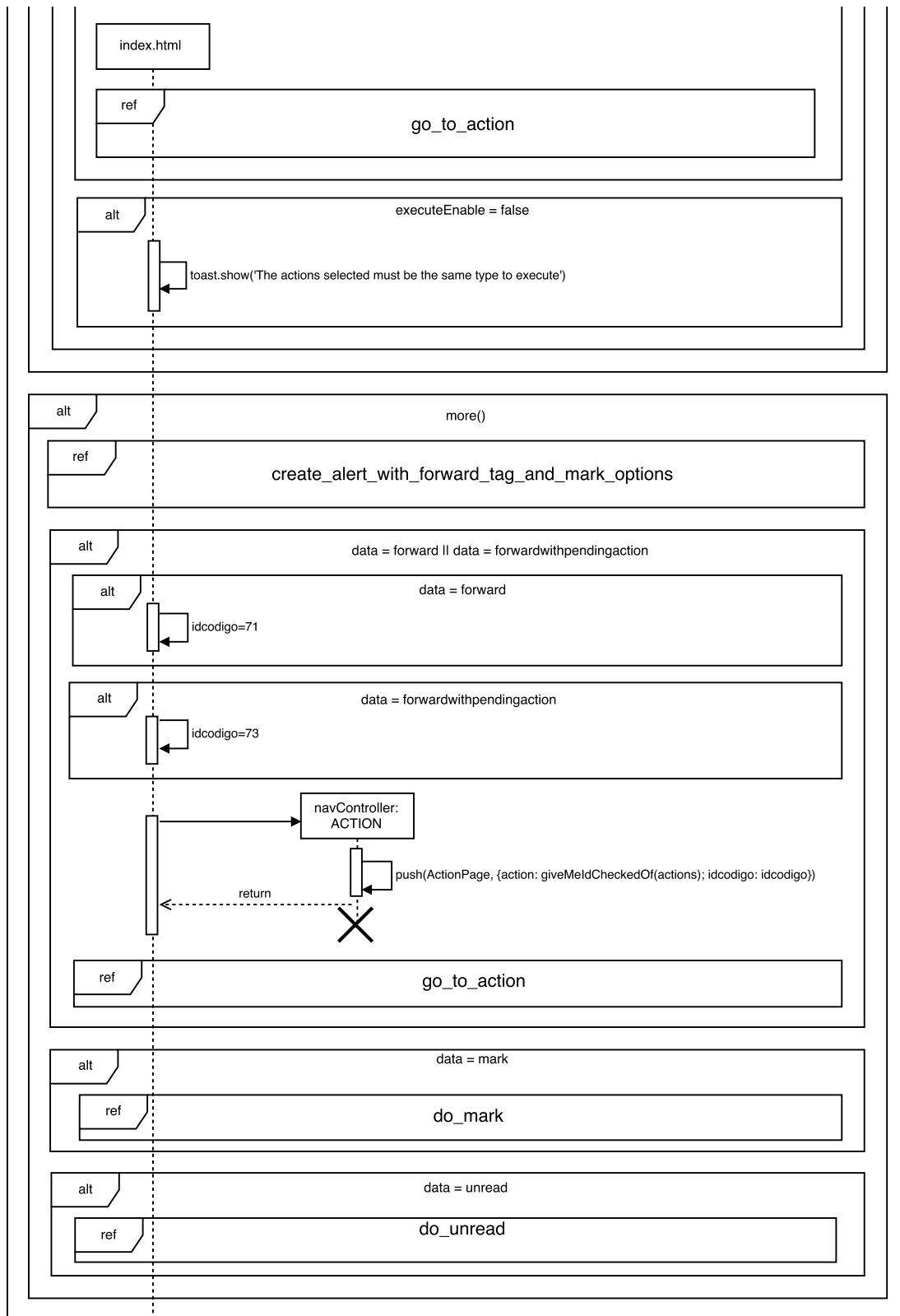


Figura B.9: Diagrama de Sequência - Lista de Ações (5 de 5)

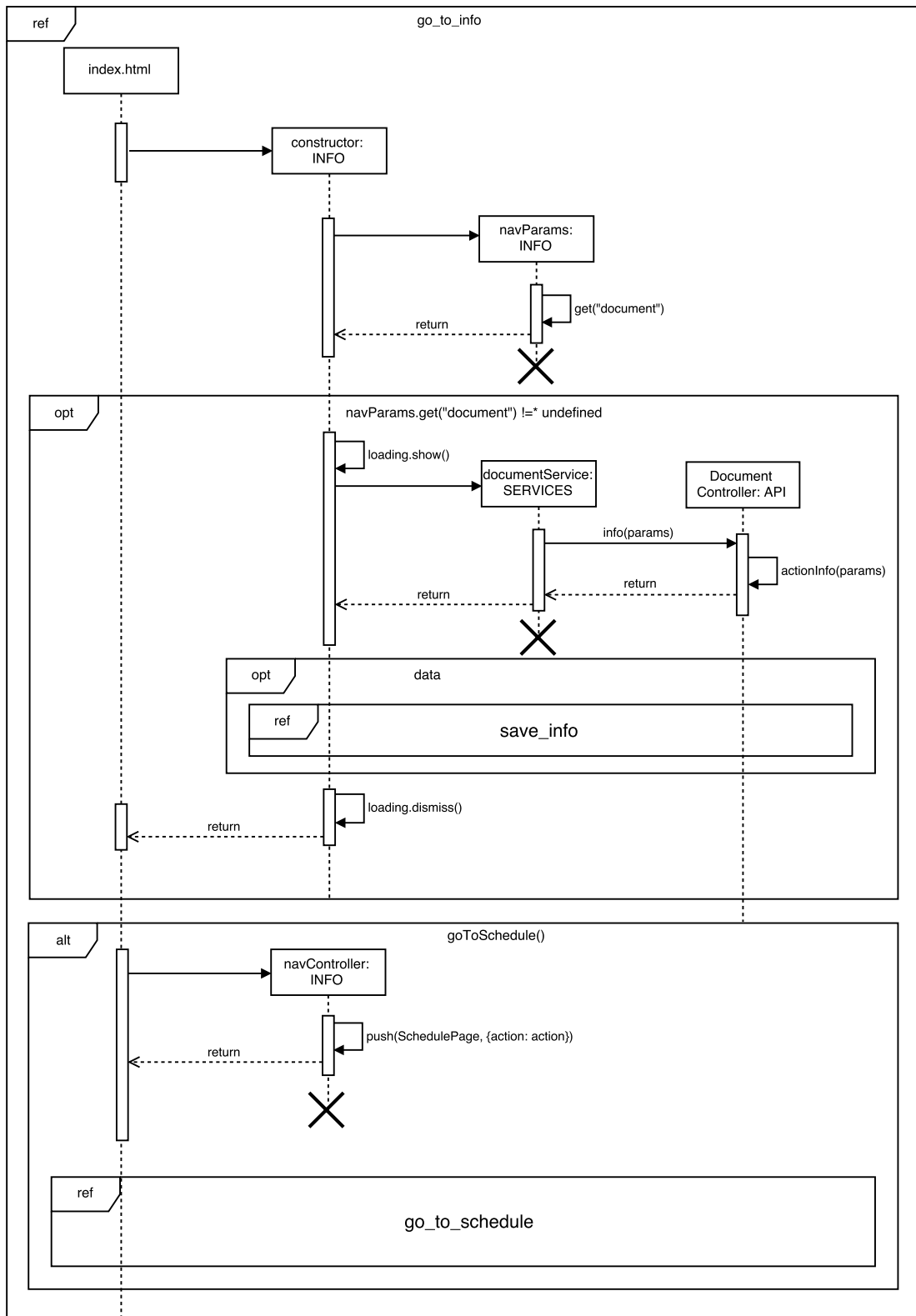


Figura B.10: Diagrama de Sequência - Info (1 de 2)

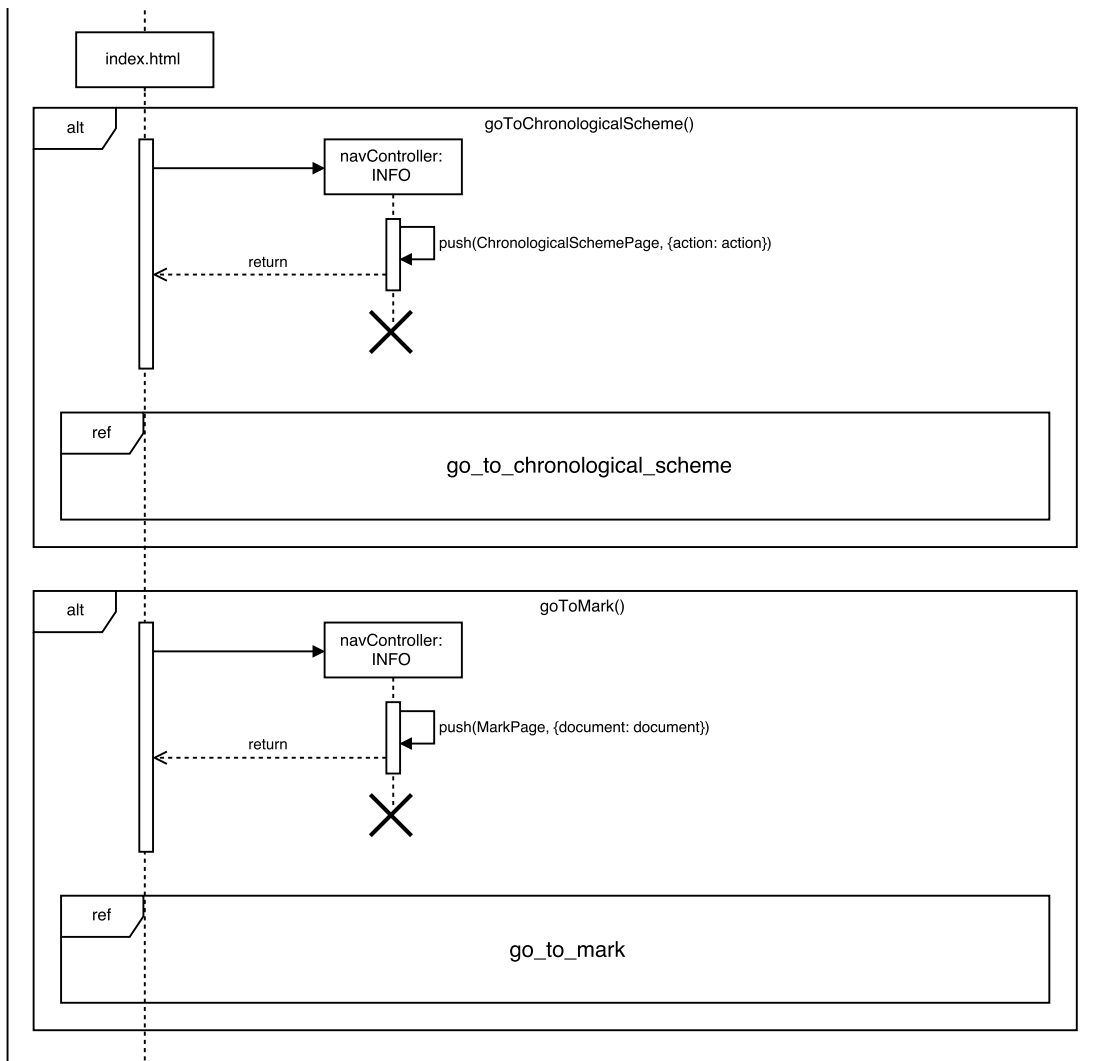


Figura B.11: Diagrama de Sequência - Info (2 de 2)

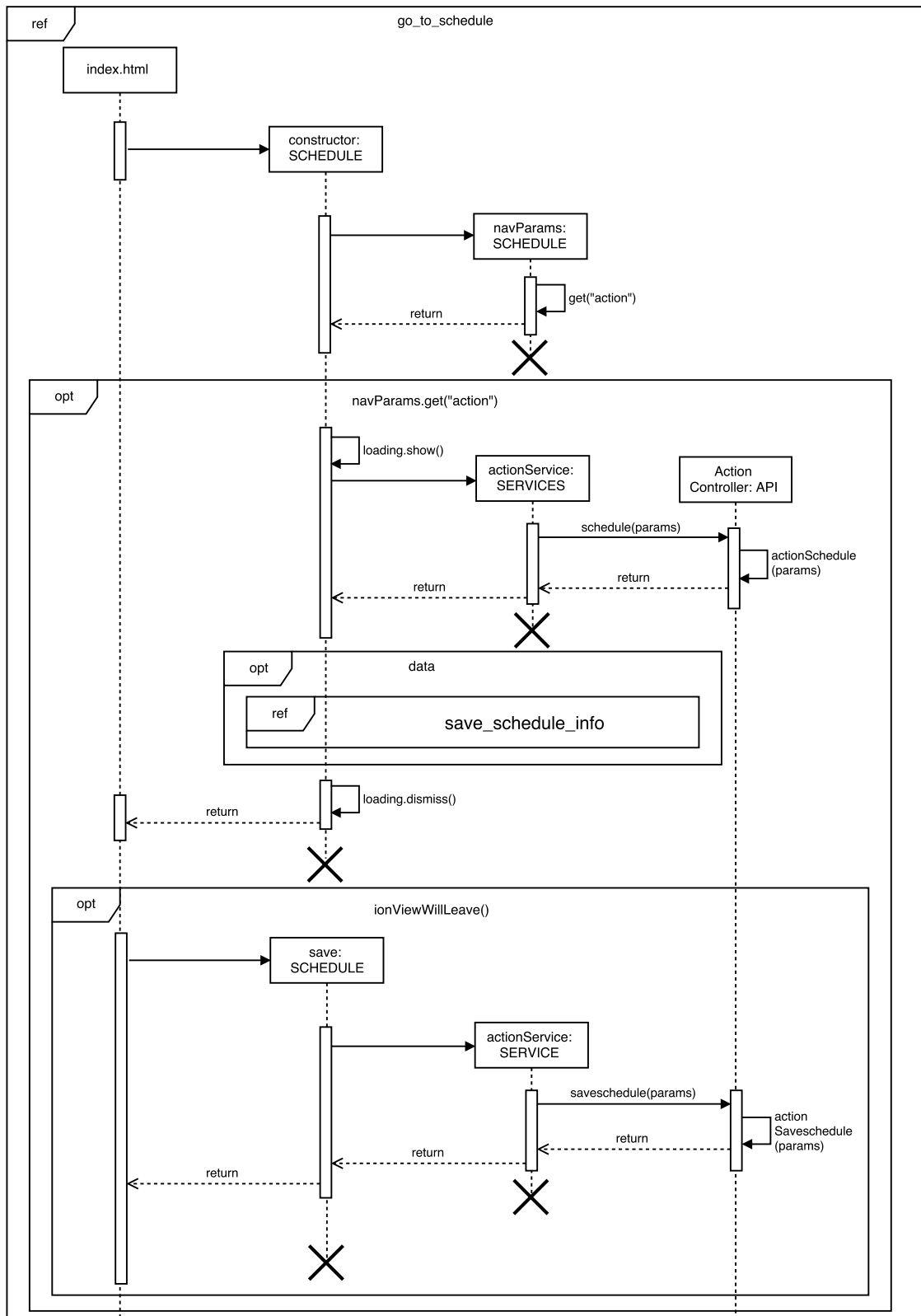


Figura B.12: Diagrama de Sequência - Schedule (1 de 2)

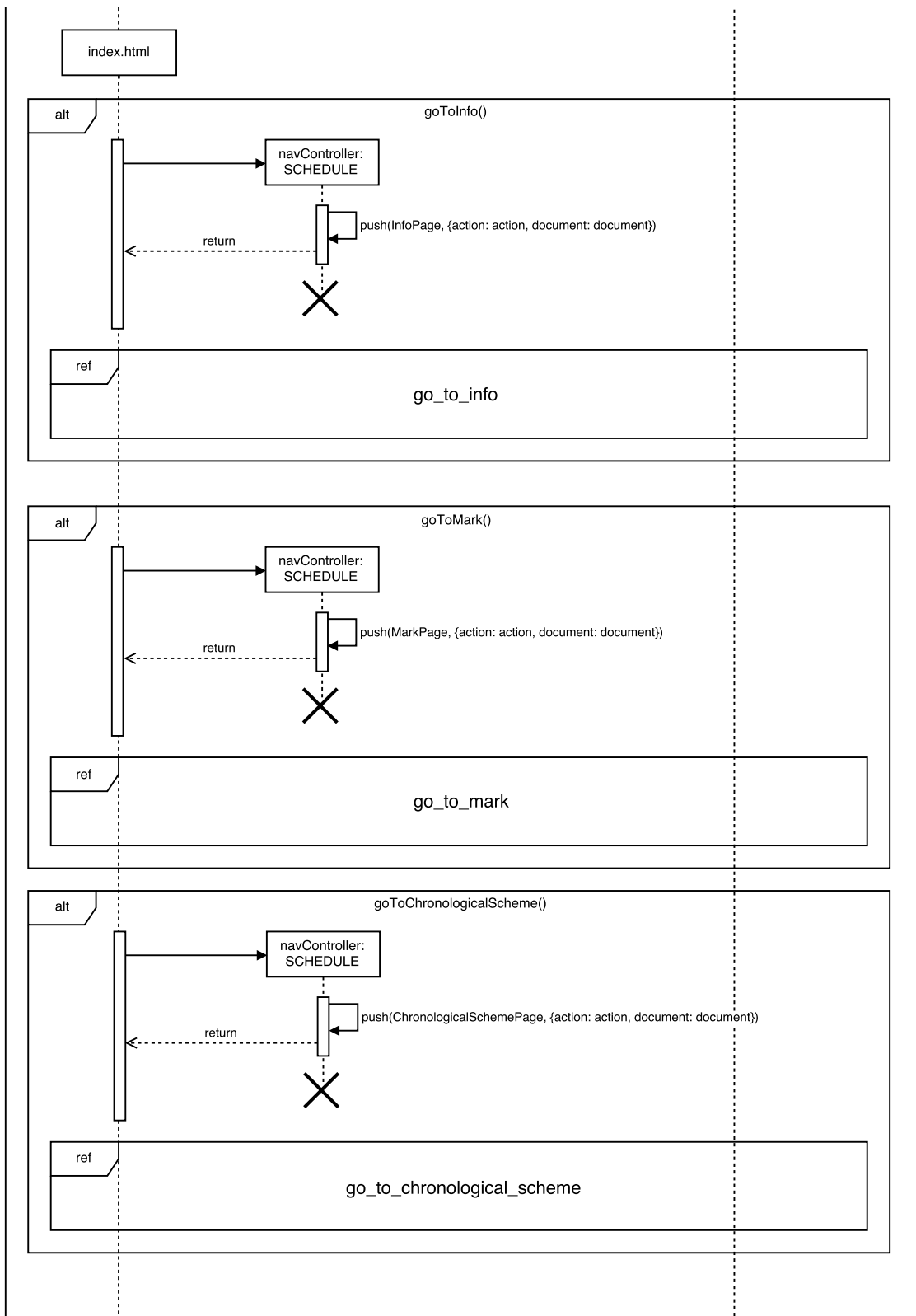


Figura B.13: Diagrama de Sequência - Schedule (2 de 2)

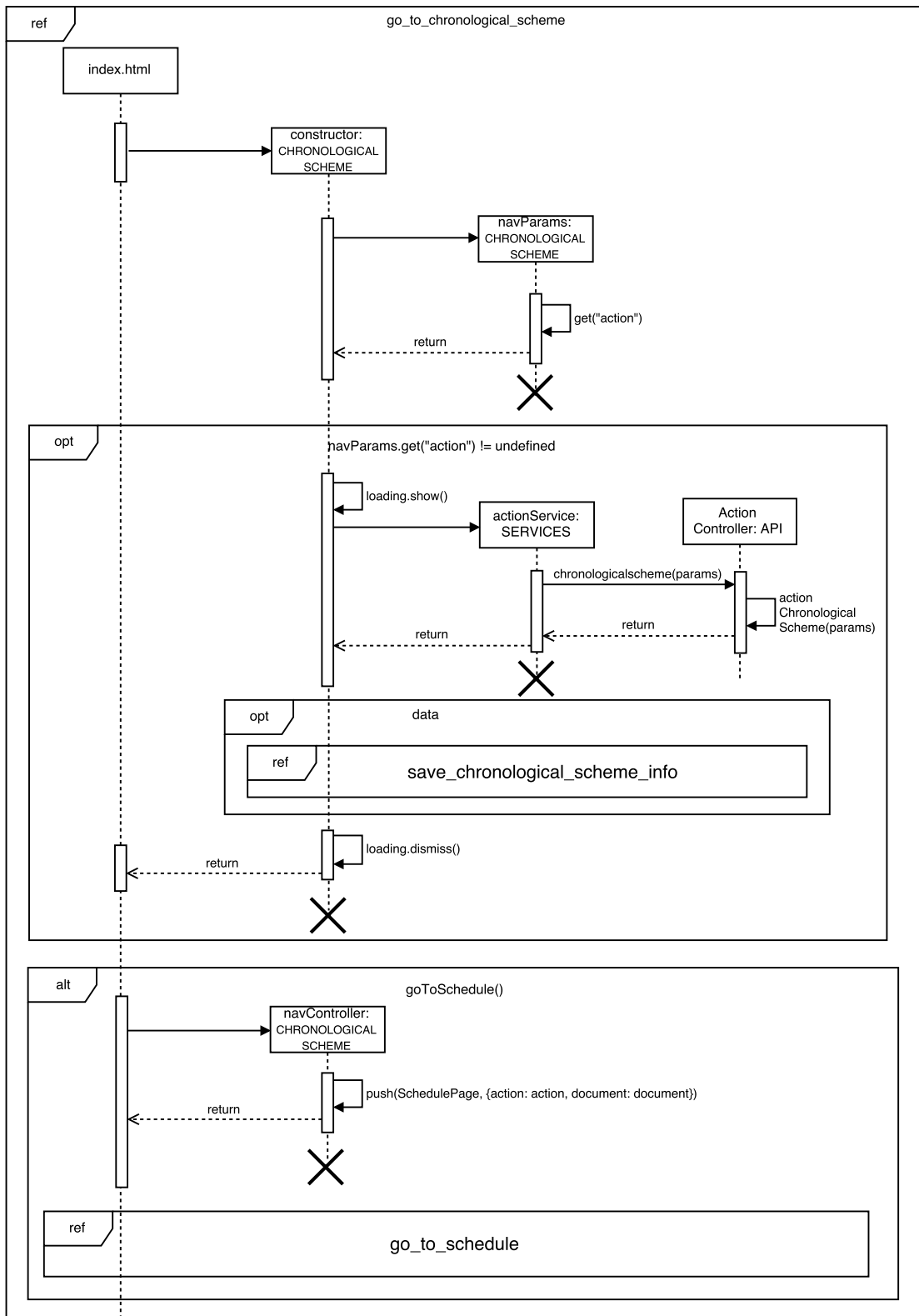


Figura B.14: Diagrama de Sequência - Chronological Scheme (1 de 2)

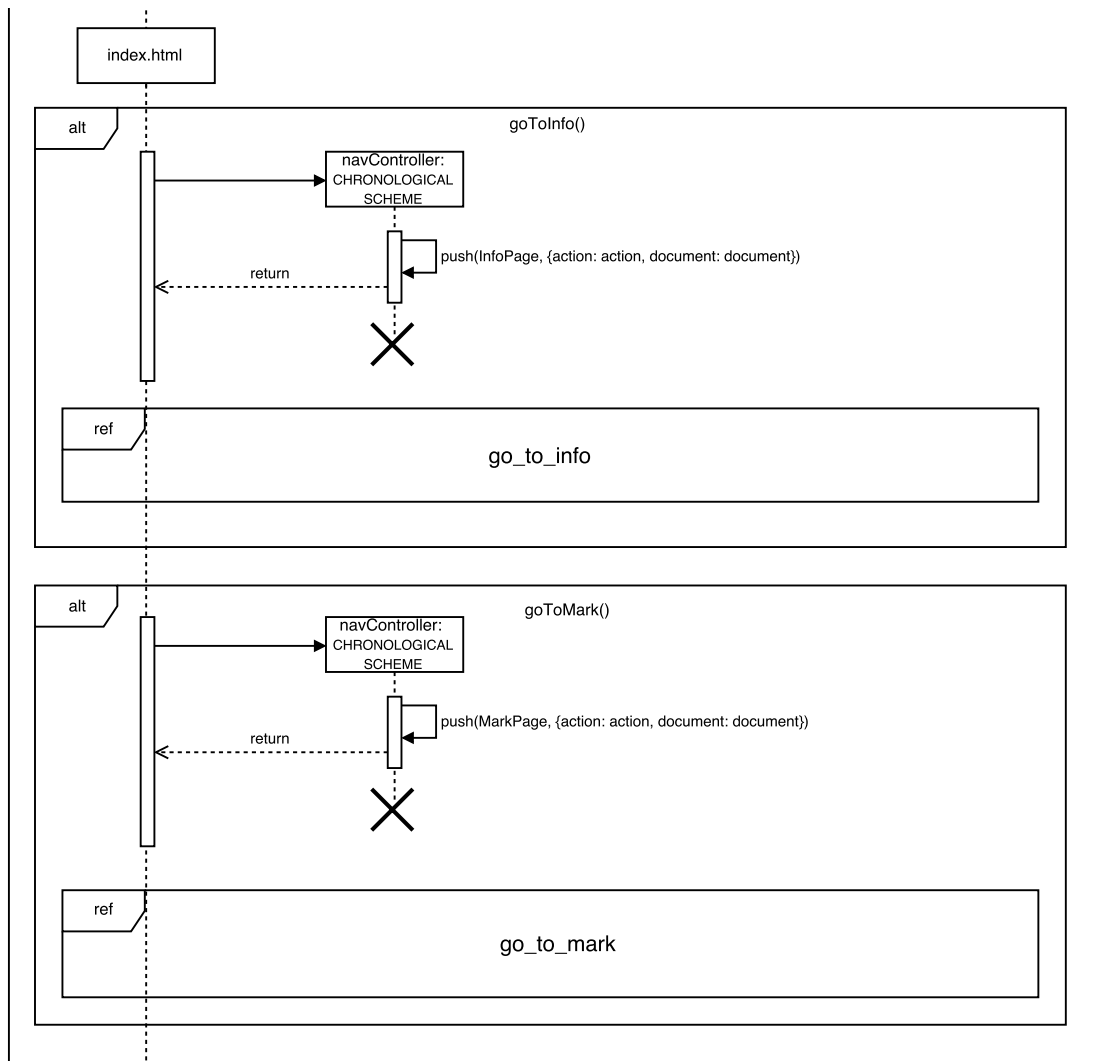


Figura B.15: Diagrama de Sequência - Chronological Scheme (2 de 2)

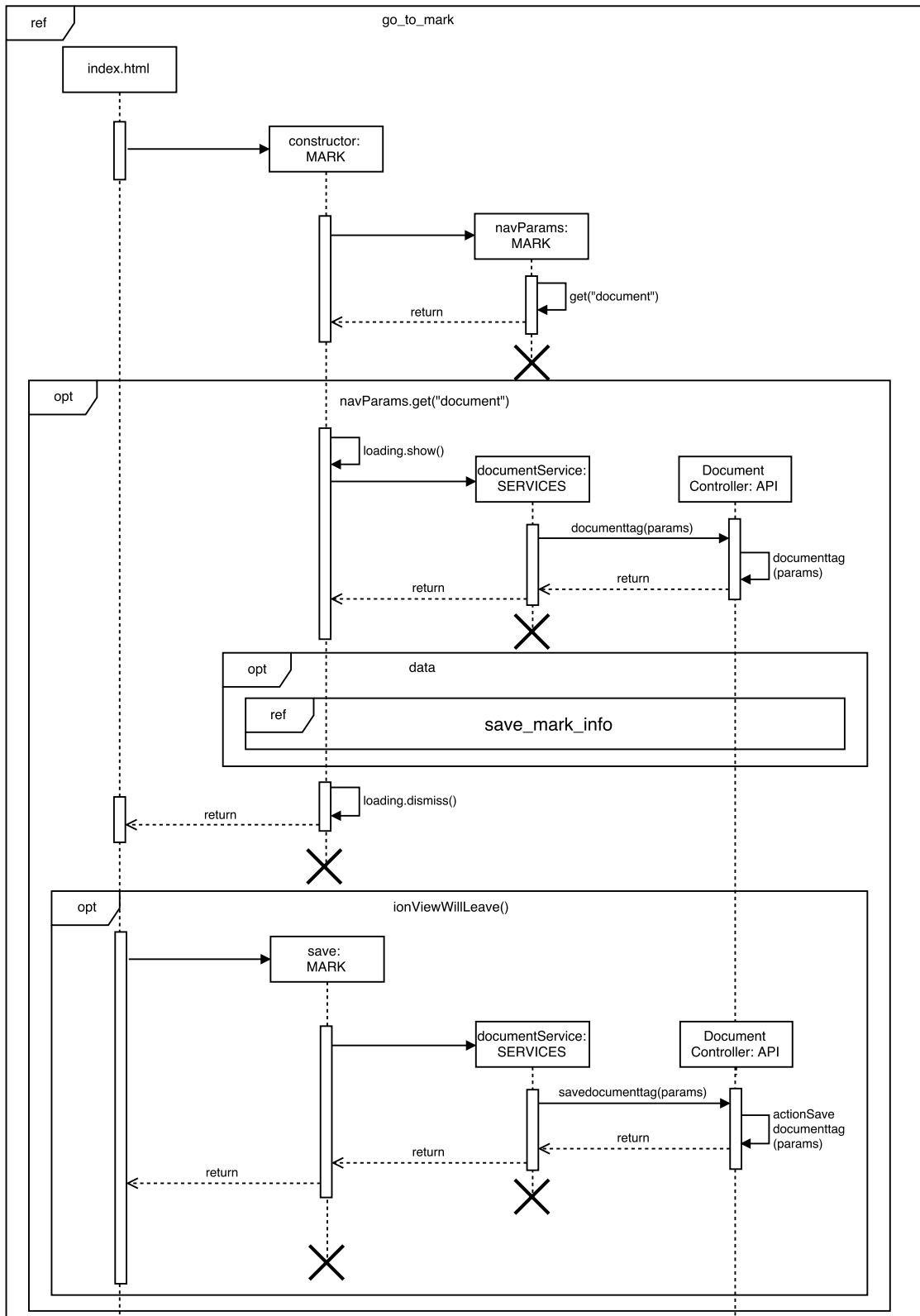


Figura B.16: Diagrama de Sequência - Mark (1 de 2)

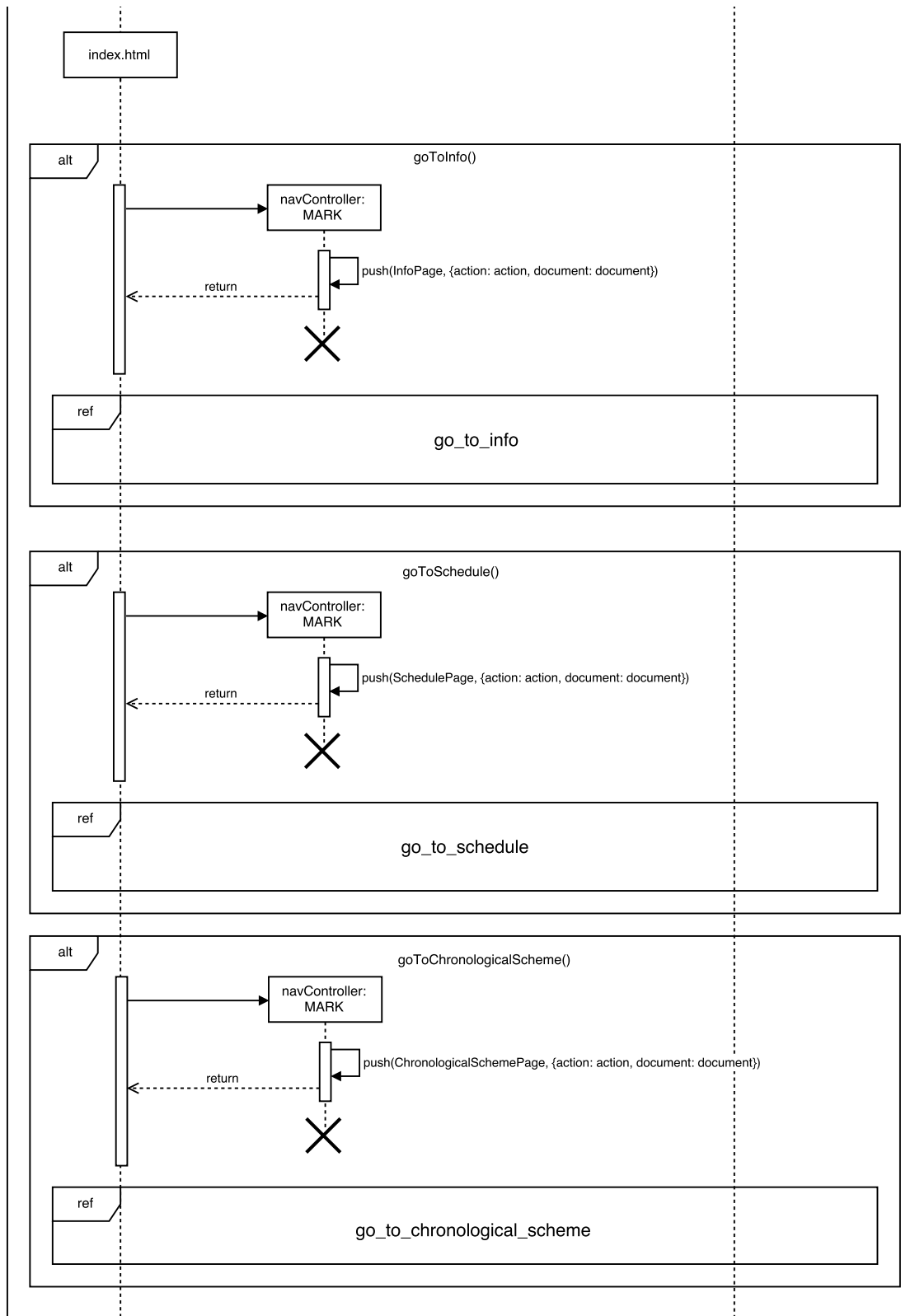


Figura B.17: Diagrama de Sequência - Mark (2 de 2)

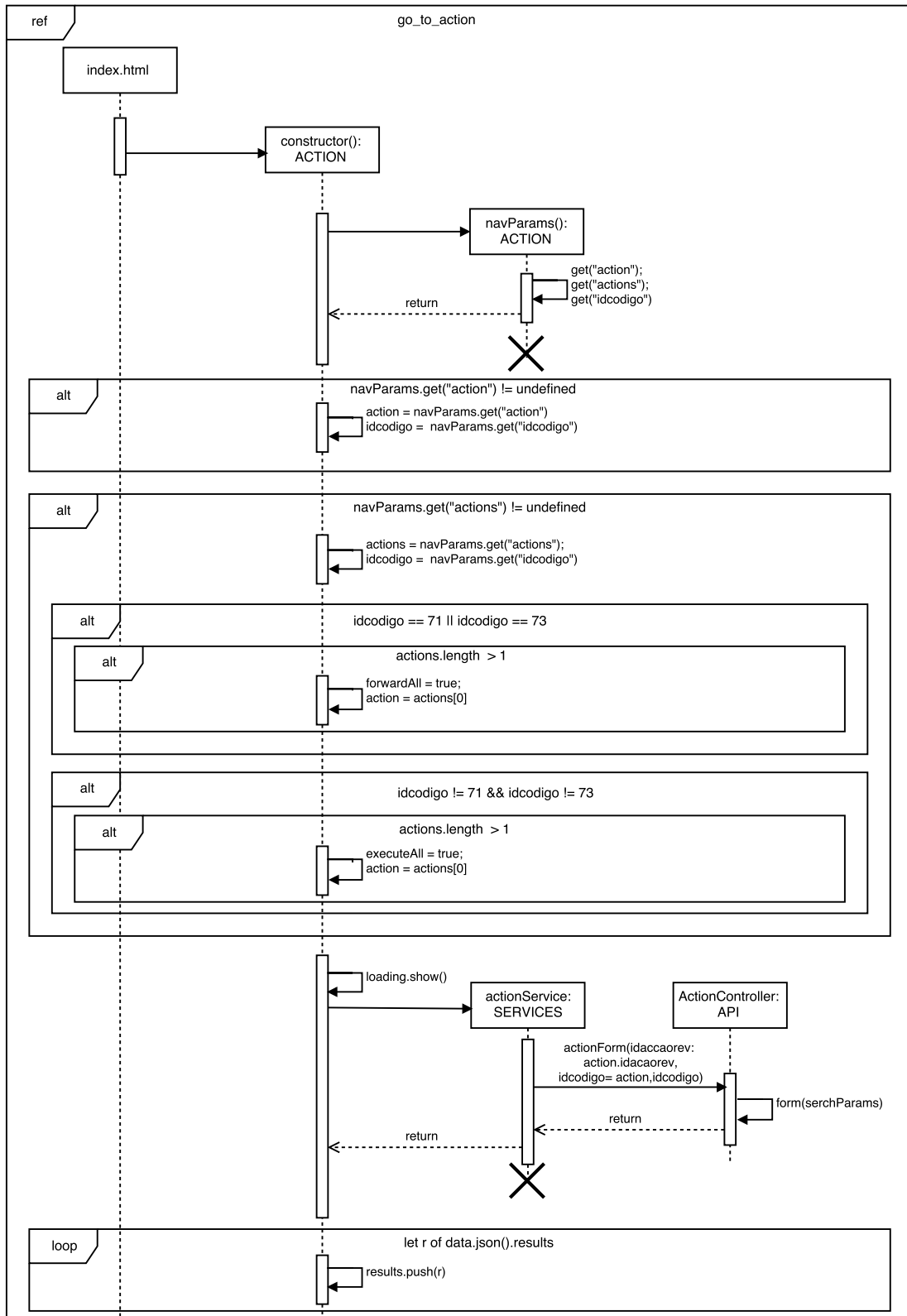


Figura B.18: Diagrama de Sequência - Action (1 de 4)

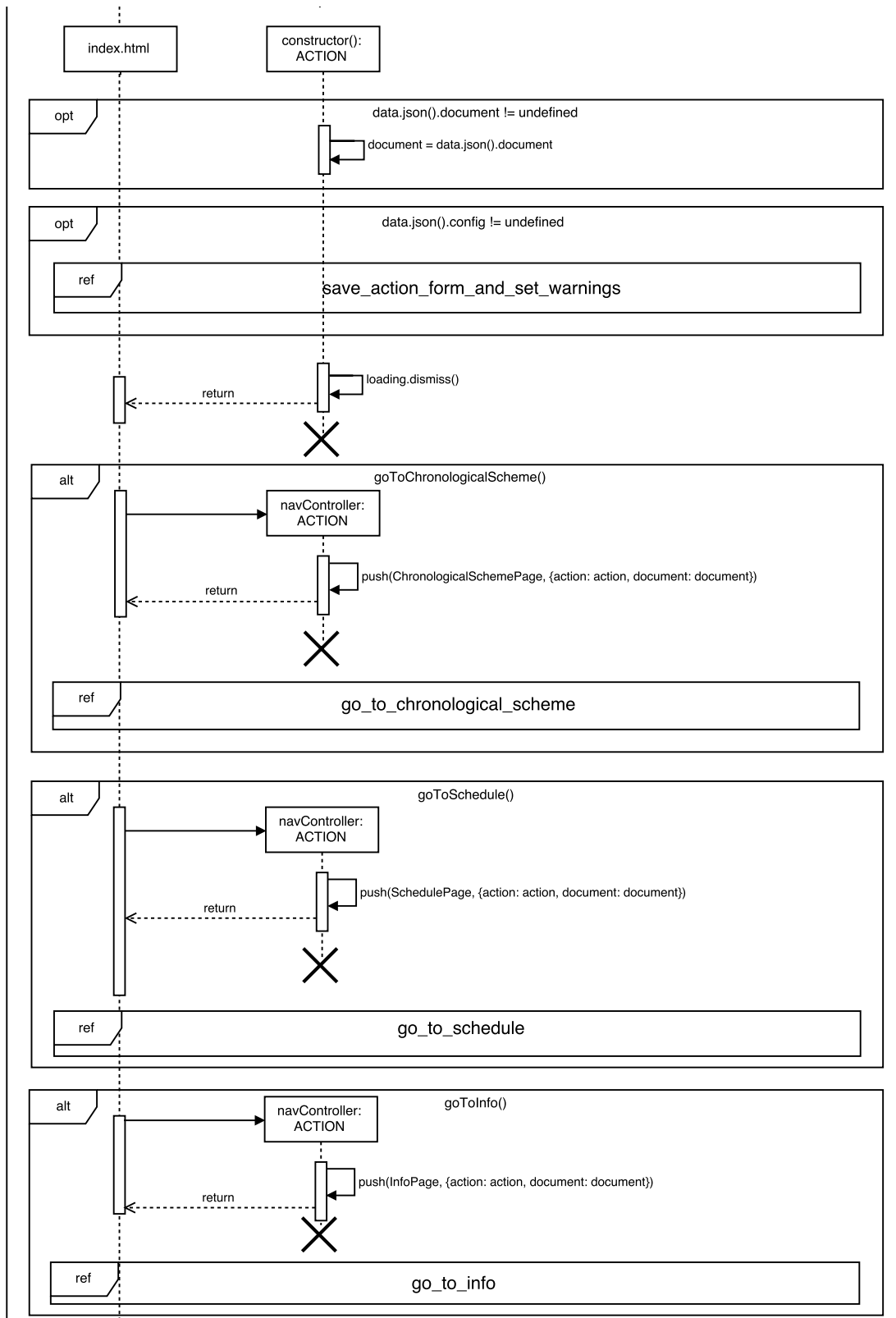


Figura B.19: Diagrama de Sequência - Action (2 de 4)

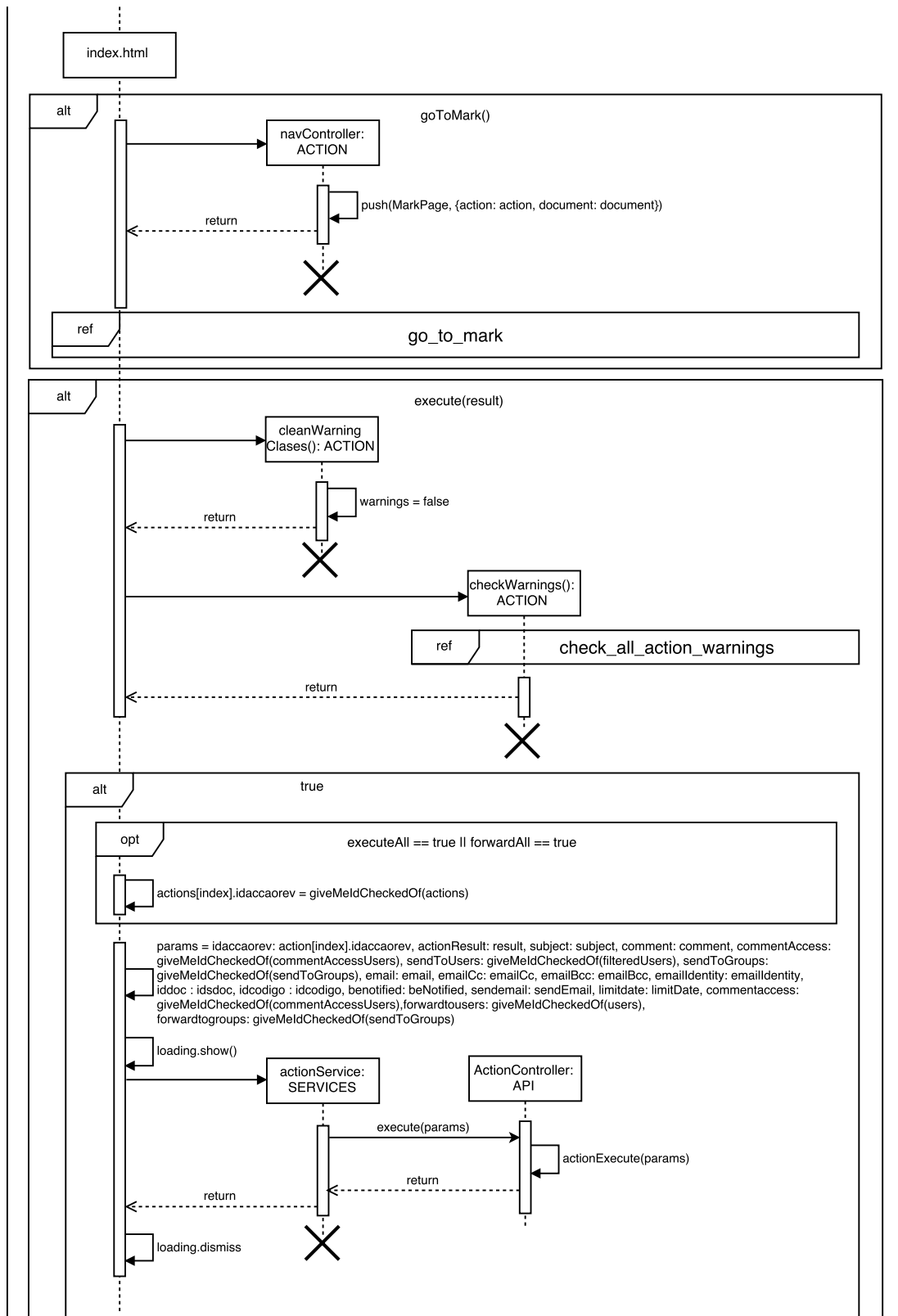


Figura B.20: Diagrama de Sequência - Action (3 de 4)

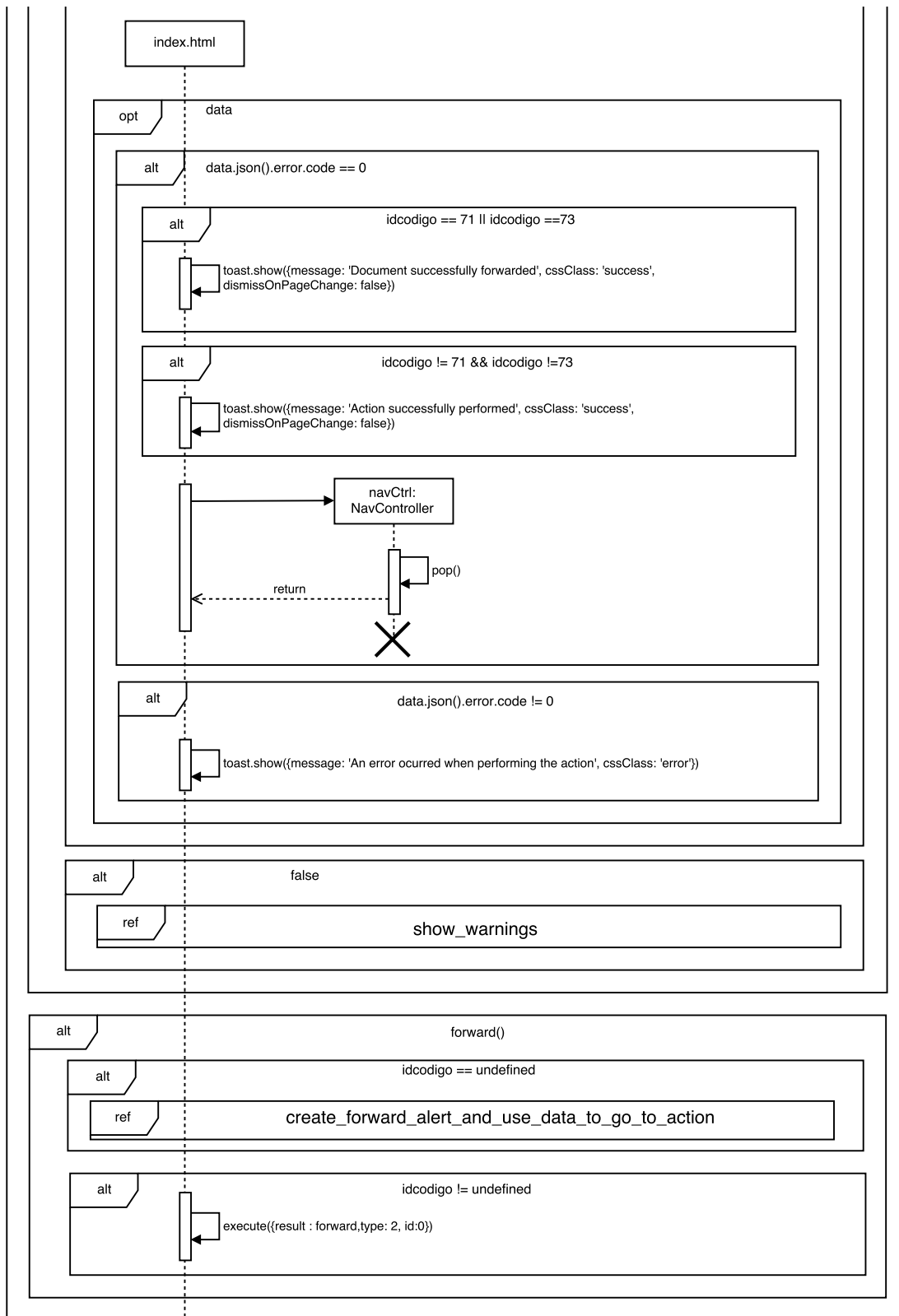


Figura B.21: Diagrama de Sequência - Action (4 de 4)

Referências

- [1] Mobile app development. <http://initiontechnology.com/mobile-app-development/>. Acedido em: 2017-01-24.
- [2] Google makes mobile optimization seo ranking factor. <http://www.socialmediatoday.com/social-business/2015-04-07/google-makes-mobile-optimization-seo-ranking-factor/>. Acedido em: 2017-01-25.
- [3] Apache cordova overview. <https://cordova.apache.org/docs/en/latest/guide/overview/>. Acedido em: 2017-02-7.
- [4] Ionic component documentation - action sheets. <http://ionicframework.com/docs/v2/components/#action-sheets>. Acedido em: 2017-02-8.
- [5] Hassenzahl M. e Roto V. Being and doing: a perspective on user experience and its measurement. *Interfaces*, 72:10–12, 2007.
- [6] L Srivastava. Mobile phones and the evolution of social behaviour. *Behaviour & Information Technology*, 24:111–129, 2005.
- [7] Cisco & C. Cisco. The cisco® visual networking index (vni) global mobile data traffic forecast update (2015 – 2020). <https://www.cisco.com/c/en/us/solutions/collateral/service-provider/visual-networking-index-vni/mobile-white-paper-c11-520862.html>. Acedido em: 2017-01-24.
- [8] IPBRICK. Gestão documental, gestão de processos e comunicações numa Única solução, disponível em <http://www.ipbrick.com/pt-pt/iportaldoc-software-gestao-documental/>.
- [9] Adream Blair-Early e Mike Zender. User interface design principles for interaction design, disponível em <https://pt.scribd.com/document/85078772/User-Interface-Design-Principles-for-Interaction-Design>. Relatório técnico, Massachusetts Institute of Technology Design Issues, Inverno 2008.
- [10] Don Norman e Jakob Nielsen. The definition of user experience (ux), disponível em <https://www.nngroup.com/articles/definition-user-experience/>.
- [11] M. Hassenzahl e N. Tractinsky. User experience - a research agenda. *Behaviour & Information Technology*, 25:91–97, 2005.
- [12] Park K., Han S.H., Kim H.K., Park W., e Cho Y. Developing elements of user experience for mobile phones and services: survey, interview, and observation approaches. *Human Factors and Ergonomics in Manufacturing and Service Industries*, 2011.

- [13] Sebastian Deterding, Dan Dixon, Rilla Khaled, e Lennart Nacke. From game design elements to gamefulness: Defining “gamification”. Relatório técnico, In Proceedings of the 15th International Academic MindTrek Conference: Envisioning Future Media Environments, Setembro 28-30 2011.
- [14] Kai Huotar e Juho Hamari. Defining gamification - a service marketing perspective. Relatório técnico, In Proceedings of the 16th International Academic MindTrek Conference, Outubro 3-5 2012.
- [15] Juho Hamari e Veikko Eranti. Framework for designing and evaluating game achievements. Relatório técnico, Proceedings of Digra 2011 Conference: Think Design Play, Setembro 14-17 2011.
- [16] S.P. Jobs, S. Forstall, G. Christie, S.O. Lemay, S. Herz, M. van Os, B. Ording, W.C. Novick, G. Westerman, e I. Chaudhri. Touch screen device, method, and graphical user interface for customizing display of content category icons , disponível em <https://www.google.com/patents/US9335924>. 2016.
- [17] March 31) Christensson, P. (2009). User interface definition, disponível em https://techterms.com/definition/user_interface.
- [18] The 10 principles of mobile interface design. <http://www.creativebloq.com/mobile/10-principles-mobile-interface-design-4122910>. Acedido em: 2017-01-14.
- [19] Ketan Anant More e Ms.Priya Chandran. Native vs hybrid apps. *International Journal of Current Trends in Engineering & Research (IJCTER)*, 2:563–572, 2016.
- [20] Frameworks: O que é um framework? <http://www.dsc.ufcg.edu.br/~jacques/cursos/map/html/frame/oque.htm>.
- [21] B. Fling. *Mobile Design and Development: Practical concepts and techniques for creating mobile sites and web apps*. Animal Guide. O’Reilly Media, 2009. URL: <https://books.google.pt/books?id=LyMeulBTkH0C>.
- [22] Pros and cons of the top 5 cross-platform tools. <https://www.developereconomics.com/pros-cons-top-5-cross-platform-tools>. Acedido em: 2017-02-7.
- [23] Bootstrap is the most popular html, css, and js framework for developing responsive, mobile first projects on the web. <https://getbootstrap.com/>. Acedido em: 2017-01-29.
- [24] IPBRICK. Php frameworks - ipbrick formação - sessão 1. Relatório técnico, IPBRICK SA, 2017.
- [25] IPBRICK. iportaldoc - document management - user manual. Relatório técnico, IPBRICK SA, 2015.
- [26] What’s the difference between the front-end and back-end? <https://www.pluralsight.com/blog/film-games/whats-difference-front-end-back-end>. Acedido em: 2017-05.
- [27] Http status messages. https://www.w3schools.com/tags/ref_httpmessages.asp. Acedido em: 2017-04.

- [28] Mobile apps with html, css & js. target multiple platforms with one code base. free and open source. <http://cordova.apache.org/>.
- [29] Build amazing native apps and progressive web apps with ionic framework and angular. <http://ionicframework.com/>. Acedido em: 2017-02-8.
- [30] Ionic component documentation. <http://ionicframework.com/docs/components/#overview>. Acedido em: 2017-02-8.
- [31] Yii php framework: Best for web 2.0 development. <http://www.yiiframework.com/>.
- [32] Mvc xerox parc 1978-79, disponível em <http://heim.ifi.uio.no/~trygver/themes/mvc/mvc-index.html>.
- [33] Dave Gandy. Examples - lots of easy ways to use font awesome - animated icons, disponível em <http://fontawesome.io/examples/#animated>.