



Gener | 2011

Interfície per al control de dispositius domòtics

Jonathan Frutos Rodrigo

Projecte Final de Carrera

DADES DEL PROJECTE

Títol del Projecte: Interfície per al control de dispositius domòtics

Nom de l'estudiant: JONATAN FRUTOS RODRIGO

Titulació: ENGINYERIA EN INFORMÀTICA

Crèdits: 37,5

Director/Ponent: CRISTIAN BARRUE / CLAUDIO ULISES CORTÉS GARCÍA

Departament: LSI

MEMBRES DEL TRIBUNAL (*nom i signatura*)

President: MIQUEL SANCHEZ MARRE

Vocal: TERESA MONREAL ARNAL

Secretari:

QUALIFICACIÓ

Qualificació numèrica:

Qualificació descriptiva:

Data:

Índex

Índex	5
1. Introducció.....	9
1.1. Breu descripció del projecte.....	9
1.2. Objectius del projecte.....	10
1.3. Estructura general del projecte.....	11
2. Propòsit	13
2.1. Objectius.....	13
2.2. Abast.....	13
2.3. Requisits.....	15
2.3.1. Requisits funcionals.....	16
2.3.2. Requisits no funcionals	18
3. Gestió del projecte	19
3.1. Fases del projecte.....	19
3.1.1. Control de gestió.....	19
3.1.2. Tasques prèvies.....	20
3.1.3. Anàlisi i disseny.....	21
3.1.4. Implementació.....	22
3.1.5. Proves.....	22
3.1.6. Finalització projecte	23
3.2. Recursos.....	24
3.2.1. Recursos requerits	24
3.2.2. Recursos disponibles.....	24
3.3. Temps.....	25
4. Anàlisi inicial	27
4.1. Tecnologies candidates.....	27
4.1.1. Java.....	27
4.1.3. Tecnologies Web.....	28
4.1.4. XUL.....	28
4.1.5. Reproducció de so	29
4.2. Tecnologies implicades.....	29
4.3. Proves tecnològiques.....	30
4.3.1. Recaptació documentació XUL.....	30
4.3.2. Proves unitàries	30
4.3.3. Proves integrals	31
5. Especificació.....	33
5.1. Model de casos d'ús.....	33
5.1.1. Actors del sistema	33
5.2. Diagrama dels casos d'ús	34
5.3. Especificació dels casos d'ús.....	36
5.3.1. Gestionar aplicació.....	36
5.3.2. Gestionar perfils.....	40
5.3.3. Gestionar recursos	43

5.3.4. Gestionar Dispositius Domòtics.....	45
5.3.5. Gestionar aplicació (client).....	45
5.4. Model de comportament.....	48
5.4.1. Gestionar aplicació.....	48
5.4.2. Gestionar perfils.....	51
5.4.3. Gestionar recursos.....	52
5.4.4. Gestionar DLA.....	53
5.4.5. Gestionar aplicació (client).....	54
5.5. Model conceptual de dades.....	56
5.5.1. Vista.....	57
5.5.2. Controlador.....	58
5.5.3. Domini.....	59
6. Disseny	61
6.1. Disseny de l'arquitectura	61
6.1.1. Vista.....	61
6.1.2. Controlador.....	61
6.1.3. Domini [JS + arxius XML + binaris].....	62
6.2. Arquitectura física del sistema.....	62
7. Disseny gràfic.....	65
7.1. Elements clau de la interfície	65
7.2. Zona de configuració.....	66
7.3. Zona de navegació.....	68
7.4. Zona de previsualització	69
8. Implementació.....	71
8.1. Directori del Sistema	71
8.2. Inicialització aplicació.....	73
8.3. Persistència de fitxers	75
8.4. Connexió DLA.....	75
8.5. Generació d'aplicacions.....	77
8.6. Exemple d'implementació	79
9. Proves.....	85
9.1. Proves unitàries	86
9.2. Proves integrals	86
9.3. Proves de regressió	86
9.4. Proves d'acceptació	87
10. Planificació del projecte	89
10.1. Planificació temporal	89
10.1.1. Diagrama de Gantt	90
10.2. Anàlisi econòmica.....	91
10.2.1. Recursos humans.....	91
10.2.2. Recursos software i hardware	93
10.2.3. Cost total	94
11. Conclusions	95

11.1. Objectius assolits	95
11.1.1. Adaptació a les necessitats de l'usuari final	95
11.1.2. Interacció amb dispositius domòtics.....	95
11.1.3. Configuració personalitzada	96
11.1.4. Vista prèvia.....	96
11.1.5. Independència de l'entorn de generació respecte al d'execució	96
11.2. Conclusions del projecte.....	97
11.3. Valoració personal	98
12. Propostes de millora	101
12.1. Comandes de veu	101
12.2. Detector de presència – generació automàtica de pantalles	101
12.3. Augment de formats suportats	102
13. Bibliografia.....	103
14. Annexos	105
14.1. Mètodes d'interacció amb una interfície gràfica.....	105
<i>Hardware</i>	105
<i>Software</i>	110
14.2. Manual d'instal·lació d'entorn	112
14.3. Manual d'ús	113
<i>Interfície d'administrador</i>	113
<i>Interfície de client (usuari final)</i>	132
14.4. Índex d'il·lustracions.....	135
14.5. Índex de taules.....	138

1. Introducció

1.1. Breu descripció del projecte

L'envelliment poblacional és un fet que ha anat cristal·litzant any rere any fins a convertir-se, segons un *estudi de la ONU realitzat durant el 2009* [veure bibliografia], en un fet sense precedents, generalitzat, profund i permanent. Això significa que cada cop hi ha més urgència en satisfer les necessitats d'un sector de la població envellit, amb tot el que comporta aquest envelliment a nivell de la salut tant física com mental.

Amb aquest motiu de rerefons, es pretén usar els diferents avenços tecnològics de que es disposa per a millorar la qualitat de vida de les persones que en l'actualitat tenen dificultats per a portar una vida independent. Es tracta no només d'aquelles persones d'avançada edat amb diferents problemes motrius o cognitius, sinó també d'aquelles persones de qualsevol edat que presenten tipologies clíniques que dificulten portar una vida independent o, més ben dit, el màxim independent factible dins de les possibilitats de cada individu.

En l'actualitat ja existeixen diferents dispositius tecnològics que permeten crear edificis i habitatges automàtics amb la finalitat de millorar la gestió energètica i la qualitat de vida dels seus habitants: reben el nom de dispositius domòtics. La domòtica s'aplica bàsicament sobre quatre àrees: Benestar (increment de confort a la llar), Estalvi (gestió del consum elèctric de l'habitatge), Seguretat (Alarmes, Presència simulada durant llargs períodes d'absència) i Comunicacions (vídeo i so distribuït per tota la llar, telefonia IP). Però tots aquests dispositius, per a gent amb limitacions funcionals, van més enllà d'un simple augment de comoditat, i es que també poden suposar una millora substancial de la qualitat de vida, complimentant aquelles capacitats que s'hagin vist minvades.

El problema que presenta la domòtica a nivell general, però, es que aquests dispositius resulten ineficaços en quant cadascun d'ells està aïllat de la resta; i aconseguir una casa capaç d'integrar de forma plena diferents elements domòtics, amb la finalitat d'ajudar a millorar el nivell de vida del usuari, pot arribar a ser una tasca complicada degut al nivell de fragmentació referent al control dels dispositius instal·lats.

Amb aquestes raons, es pretén realitzar un estudi i posterior desenvolupament d'un sistema prototípic que integri un ventall de dispositius domòtics actuals dins d'un mateix embolcall de control, sobre el que dissenyar una interfície gràfica que permeti adaptar-se tant a possibles dificultats motrius com cognitives del usuari; premiant la cerca d'un sistema usable per damunt d'altres

aspectes, doncs es pretén maximitzar la quantitat de públic capacitat per a usar la interfície gràfica de comunicació amb els diferents dispositius domòtics.

Una tasca d'aquestes característiques no es pot dur a terme de forma plenament individual, ja que es necessiten infraestructures i equips domòtics per dur a terme totes aquelles proves que es considerin necessàries; mentre que els coneixements sobre conducta humana també han de ser coberts d'alguna forma per experts en la matèria, doncs s'escapen de l'àmbit tecnològic.

Es per aquest motiu que el projecte es realitza conjuntament amb el Centre de Vida Independent (d'ara endavant, CVI), un referent català en la promoció de l'autonomia personal orientat a la recerca, desenvolupament, innovació i validació de productes de suport i tecnologia aplicada a les persones dependents; on des d'aquest centre es subministra al projecte amb els recursos materials (en forma de dispositius domòtics) i humans (psicòlegs especialitzats en les àrees de deficiència motriu i/o cognitiva).

Aquesta iniciativa sorgeix, per tant, com un projecte presentat pel CVI i la UPC a la ONCE per desenvolupar una interfície gràfica capaç d'interaccionar amb diferents dispositius domòtics, tot adaptant-se a les necessitats cognitives i motrius de la població objectiu.

1.2. Objectius del projecte

El principal objectiu d'aquest projecte es realitzar una interfície per a dispositius domòtics que sigui adaptable. Concretament es tracta de determinar les necessitats d'un sector de la població amb problemes de dependència i establir una interfície altament adaptable que es comuniqui amb diferents elements domòtics de la llar per a facilitar i millorar fins el possible les condicions de vida d'aquest sector de la població.

Podem concloure, per tant, que els objectius marcats per aquest projecte són els següents:

- Ampliar el coneixement sobre els aspectes d'usabilitat i adaptabilitat pel que respecta a interfícies.
- Determinar els requisits necessaris per al col·lectiu de gent amb problemes d'independència.
- Crear una interfície gràfica que es comuniqui amb diferents elements domòtics.

Pel que fa al mètode de comunicació amb els diferents elements domòtics, es disposarà com a mètode d'enllaç entre la interfície gràfica i els dispositius domòtics de la DLA. La DLA [E.J. Pérez: "Arquitectura de Navegación Distribuida para Agentes Robóticos", Tesis Doctoral, Servicio de Publicaciones de la Universidad de Málaga, ISBN: 84-690-3326-3, 2006] és una arquitectura que respon a les inicials "Distributed Layer Architecture" i s'utilitza per implementar sistemes cooperatius que necessiten la interacció de diferents mòduls distribuïts lliurement en diferents màquines.

Aquest projecte es basarà exclusivament en usuaris pertanyents al col·lectiu de persones amb algun tipus de dificultat cognitiva o motriu, ja que es pretén millorar el seu nivell de vida.

1.3. Estructura general del projecte

El projecte s'inicia amb una introducció a quins són els objectius a assolir, diferenciant entre el que s'entén que forma part del projecte i delimitant allò que es considera fora del mateix (l'abast).

Tot seguit es realitzen la detecció de requisits del projecte i un anàlisi general del mateix que permetrà dividir el projecte en trossos més petits que facilitaran resoldre un problema de grans dimensions mitjançant la concatenació de la resolució de problemes més petits. Aquest anàlisi aporta una perspectiva a les diferents fases d'un projecte informàtic, els recursos que es necessiten i els que realment es tenen i una visió de l'acotament temporal al que es troba subjecte el projecte.

El seguiran la definició de cadascun dels problemes a solucionar des d'un de vista tècnic, acompanyants del diagrama corresponent. Es tracta d'especificar cada cas en el que es contempla un ús únic del sistema, a mode de recrear el comportament entre l'usuari i la interfície. Aquesta definició del cas d'ús ve acompanyada del corresponent diagrama de seqüència, o dit d'un altra manera, a cada definició d'interacció de l'usuari amb la interfície, se l'acompanya d'un diagrama explicatiu.

A continuació, un cop es té una idea de com funciona el sistema, es realitzaran els dissenys corresponents a la interfície gràfica definitiva, juntament amb els dissenys d'estructures de dades pertinents, i s'efectuarà la programació del sistema acordat. Aquest disseny gràfic és fruit d'un estudi anterior en el que s'han proposat varies interfícies gràfiques, acompanyades dels respectius anàlisis. Encara que les propostes inicials no es troben detallades en el document, aquestes han permès decidir quina proposta s'adapta millor a les

necessitats detectades oferint un grau més alt de flexibilitat i usabilitat, i les conclusions que se'n deriven de les propostes estan integrades en el disseny final.

Acostant-nos a la conclusió, es procedirà a explicar el procés d'implementació del sistema. Es pretén mostrar aquelles etapes o agrupacions de funcionalitats del procés de la implantació que presenten alguna particularitat; i com s'implanta, a mode d'exemple, una funcionalitat genèrica qualsevol. Per a la fase de comprovació, s'empraran tècniques d'assaig per analitzar el comportament de l'aplicació i la reacció dels usuaris al usar la mateixa, detectar carències en el sistema i ampliar alhora els requisits detectats en les fases inicials. Amb els resultat de les proves es farà un anàlisi detallat de les interaccions de l'usuari amb el sistema i s'extrauran conclusions. D'aquestes conclusions i de les obtingudes en l'anàlisi previ als tests, es proposaran millores del sistema per a futures implementacions.

Per finalitzar, es realitzarà un anàlisi econòmic del projecte, doncs ja es podran estimar costos materials i temporals, i es perfilaran les conclusions finals del mateix per intentar resumir tota la realització. De la mateixa manera, també s'inclouran conclusions personals del projecte amb propostes de millora si algun dia es pretén continuar el projecte.

2. Propòsit

Amb el projecte descrit, cal definir els límits que envolten una tasca com la que aquí es proposa, doncs tot projecte ha d'estar regulat per unes fites mesurables, dins d'un marc que enquadri les possibilitats a contemplar pel mateix (allò que queda definit com a responsabilitat del projecte i el que no ho és) i, finalment, definir cadascuna de les necessitats detectades que s'han de satisfer per a la consecució dels objectius del projecte.

Tots aquests passos són, de forma esquemàtica, el que es trobarà explicat a continuació en aquest capítol.

2.1. Objectius

Per entendre millor els reptes que planteja el projecte, cal establir unes regles de joc bàsiques que serveixin per definir les fites que el projecte pretén assolir i que serveixin alhora per a mesurar l'èxit o no del mateix.

- Contribuir en la millora de la qualitat de vida d'aquelles persones amb discapacitats motrius (com per exemple aquells que no es poden moure del llit), discapacitats cognitives (provocades per la malaltia d'Alzheimer per exemple) o barreges dels dos tipus de discapacitats.
- Crear una interfície gràfica capaç de controlar el sistema domòtic d'una casa, adaptant-se el millor possible a les necessitats peculiars dels usuaris.
- Crear un sistema que no depengui del Sistema Operatiu i que sigui altament modificable.

2.2. Abast

El disseny i implementació del software necessari per a una interfície que s'adaptés a les necessitats del col·lectiu d'usuaris representat pel CVI ha sigut el desencadenant de l'abast d'aquest projecte. D'aquesta forma, s'han desenvolupat i estudiat els punts o àrees següents:

- Estudi de tecnologies aplicables existents (les més significatives).
- Realització del disseny de distribució d'informació i assignació de rols a cadascuna de les tecnologies implicades.

- Realització del disseny de les interfícies a usar per part dels administradors del sistema.
- Realització del disseny d'interfície a usar per part dels usuaris del sistema.
- Realització disseny de funcionalitats per comunicar la interfície amb la API d'accés als sistemes domòtics.
- Estudi implantació intèrpret de veu.
- Implantació del sistema en equips al CVI.
- Pressupost.

Com a punts complementaris del projecte s'han desenvolupat els següents temes:

- Estudi de temps i planificació del projecte.
- Estudi accessibilitat.

Els punts que queden fora de l'abast del projecte, i per tant no es realitzaran, són:

- Realització intèrpret de comandes de veu.
- Implantació del sistema en equips d'usuari final.
- Estudi i disseny del hardware necessari.

2.3. Requisites

En aquesta secció es presenta l'anàlisi de requisits per a la interfície dels dispositius domòtics que s'implementarà per al centre CVI.

L'anàlisi de requisits compren totes les tasques relacionades amb la determinació de les necessitats o condicions a satisfer per un software, tenint en consideració els diversos requisits que aportin figures externes com ho són els inversors, etc.; podent entrar en conflicte entre ells.

Els mètodes emprats per a l'obtenció dels requisits han sigut:

- Entrevistes:
Es tracta d'un mètode comú i està quan es tracta d'obtenir requisits i validar sistemes en general. S'ha seleccionat un nombre significatiu de persones que ha representat el col·lectiu crític de persones implicades en el projecte. Aquest col·lectiu ha inclòs enginyers en Informàtica que treballen o estan directament relacionats amb el centre CVI, que han aportat els coneixements sobre les infraestructures disponibles en la casa domòtica, així com també han servit de pont per a parts crucials del projecte, com la comunicació amb els creadors de l'API de comunicació. També s'han inclòs psicòlogues expertes en l'àrea cognitiva i motriu, que han aportat el coneixement adquirit sobre el tipus d'usuaris que han de fer ús de la interfície i han esdevingut el focus real de recaptació de requisits pel projecte.
- Prototipatge:
En base a una petita mostra, de funcionalitat limitada, de com serà el producte final. Permet conèixer de primera mà l'opinió dels usuaris, rectificar alguns dels aspectes i descobrir requisits ocults a primera instància.
- Forma de contracte:
Algunes de les funcionalitats que ha de realitzar el sistema no han sigut fixades per cap demanda del grup d'usuaris ni s'han detectat durant la fase de prototipatge, sinó que han vingut preestablertes per la mateixa definició del PFC en el moment d'establir el mateix contracte. Es podrien classificar, d'alguna forma, com prerequisits del projecte i venen donats directament per la UPC en la seva posició de medidora entre el CVI i l'alumne encarregat del projecte.

Amb els mètodes d'obtenció de requisits usats, s'han obtingut una sèrie de requisits que, classificats segons la tipologia tradicional, queden exposats de la següent forma:

2.3.1. Requisits funcionals

Són tots aquells requisits que especifiquen una funcionalitat que ha de realitzar el sistema o component. A continuació es mostra un llistat amb tots els que s'han identificat:

- Flexibilitzar comportament de pantalles
 - Una pantalla ha de poder definir el nombre d'elements que s'han de mostrar a l'usuari final.
 - S'han de poder assignar els diferents dispositius de la casa de forma independent.
 - Cada pantalla ha de poder relacionar-se amb altres pantalles.
 - Els elements de cada pantalla s'han de poder mostrar en forma pictogràfica i/o textual.

- Enregistrar dispositius
 - S'ha de permetre relacionar accions de la interfície amb els diferents dispositius domòtics de la casa.

- Modificar comportament botó
 - Cada element de la pantalla de l'usuari final s'ha de poder configurar per assignar un text, imatge so i acció únics.

- Gestionar dispositius domòtics
 - S'ha de permetre obtenir un llistat amb tots els dispositius domòtics associats a la casa i les accions que cadascun d'ells pot dur a terme.
 - S'ha de permetre que cadascun dels dispositius es pugui associar/des associar a elements (botons) de la pantalla.
 - S'ha de permetre que des dels elements de pantalla es pugui accionar dispositius domòtics.

- Gestionar perfils
 - Les configuracions que es facin de la interfície s'han de poder carregar, eliminar i editar.

- Adaptar pantalla a usuaris amb agudes dificultats motrius
 - S'ha de permetre que els elements de la pantalla interactuïn amb persones amb capacitat motriu reduïda. Això inclou (però no es limita a) persones amb:
 - Espasmes musculars, les quals no poden usar sistemes tradicionals (com un ratolí).
 - Distròfia muscular i per tant incapacitat parcial o total de moure una extremitat.

- Adaptar pantalla a usuaris amb agudes dificultats cognitives
 - S'ha de permetre adaptar la interfície a aquelles persones amb algun tipus de deficiència cognitiva. S'inclouen (però no es limita a) els següents casos:
 - Persones amb una capacitat intel·lectual disminuïda.
 - Persones amb una alta deficiència visual.

- Introduir sons en el sistema
 - Donat que es requereix que el sistema reproduïxi sons, s'ha de permetre introduir i eliminar arxius de sons en el sistema.

- Introduir imatges en el sistema
 - Donat que es requereix que el sistema mostri imatges, s'ha de permetre introduir i eliminar arxius d'imatge en el sistema.

- Llistar totes les pantalles d'una configuració
 - El sistema ha de facilitar la configuració de cadascuna de les pantalles que tindrà disponibles l'usuari final.

- Vista prèvia
 - La ràpida visualització de la configuració de la pantalla es un element crucial alhora d'estalviar temps quan s'està configurant una pantalla per un usuari, i al mateix temps també permet comprovar si les configuracions de pantalla i botons son correctes ja que es presenta una pantalla exactament igual que la del client final, amb les mateix funcionalitats i consum de recursos.

- Generar aplicació usuari
 - Un cop configurada la pantalla de forma que s'acomodi a les necessitats motrius i cognitives de l'usuari final, s'ha de permetre generar una aplicació auto-continguda amb el perfil desitjat. Aquesta aplicació ha de contenir el nombre mínim i necessari de recursos per a efectes d'un funcionament òptim de l'aplicació d'usuari; per tant, implica una reducció de la mida de l'aplicació al reduir el nombre de recursos que ha de tractar .

- Canviar els atributs d'un botó
 - Cadascun dels diferents botons d'una pantalla s'ha de poder associar a un text, imatge i sons únics, així com associar-lo a una acció que pot ser de dos tipus:

- Generació d'una nova pantalla: Al clicar l'element en qüestió, aquest mostrarà una pantalla nova, amb nous elements.
- Realitzar acció: Al seleccionar l'element en qüestió, es cridarà al dispositiu domòtic associat a l'element i s'executarà l'acció sol·licitada.

2.3.2. Requisits no funcionals

Aquests requisits no fan referència a funcionalitats concretes del sistema software, sinó a les qualitats que ha de complir el mateix i que han de permetre avaluar-lo.

- Usabilitat
 - És la capacitat del programa per a ser entès, après i usat pels usuaris finals. Es tracta d'uns dels requisits no funcionals més importants, ja que els usuaris del sistema no són experts; es a dir, no es pot assumir que la majoria d'usuaris que usen el sistema tenen coneixements amplis del mateix. La interfície ha de ser clara i fàcil d'usar, i per a realitzar qualsevol acció no s'ha de superar un màxim de 3 pulsacions de ratolí.
 - Es buscarà que el llenguatge emprat en la interfície sigui el més entenedor possible i que els missatges que s'hagin de mostrar els pugui entendre un usuari no expert.

- Manteniment
 - El sistema haurà d'estar degudament documentat i el codi haurà de contenir comentaris explicatius del que fa cada part del programa per afavorir que altres persones el puguin modificar i ampliar en el futur.
 - El llenguatge de programació ha de permetre, partint del executable, editar qualsevol component del mateix.

- Visualitzar a pantalla completa
 - Aquesta aplicació no ha d'estar limitada a cap Sistema Operatiu d'escriptori en concret, per tant, s'ha d'oferir la possibilitat de generar l'aplicació tant en Windows, Linux com Macintosh.

- Compatibilitat
 - La interfície ha de mostrar el mateix comportament independentment del sistema operatiu que l'executi.

3. Gestió del projecte

Per poder saber si un projecte es viable o no, és necessari fer una projecció de tots els elements que participen d'alguna forma en el mateix. S'han de tenir en consideració quins recursos es necessitaran en cadascuna de les diferents fases del projecte, quin temps estimat i límit es té, quins riscos poden sorgir i quines alternatives o solucions són viables en cada cas.

Així doncs, una bona planificació és vital per fer una estimació realista de la vida del projecte, doncs permet preveure problemes, prendre decisions amb coneixement de causa i estimar conseqüències quan s'afronten problemes; i en aquest projecte, on alguns recursos es troben limitats amb l'ús en el temps, es especialment important, ja que també s'ha hagut de compaginar el PFC amb la feina i altres responsabilitats alienes.

Quan s'ha de planificar un projecte, és important granular cadascuna de les tasques a realitzar. Quantes més fases es puguin fer millor, ja que això permet gestionar millors el temps i recursos que són necessaris per a cada fase, i així es pot valorar millor el temps de vida necessari del projecte i detectar riscos i solucions; tot i que també s'ha de buscar un equilibri entre no granular (poc precís) i granular massa (massa feina, pot arribar a ser contraproductiu ja que s'han de gestionar masses casos i dóna lloc a inflexibilitats en el projecte).

Així doncs, el projecte s'ha dividit en les següents fases:

- Control de gestió
- Tasques prèvies (Anàlisi inicial)
- Anàlisi i disseny
- Implementació
- Proves
- Finalització (Planificació del projecte i conclusions)

Tot seguit s'explicaran amb més detall cadascuna de les fases descrites.

3.1. Fases del projecte

3.1.1. Control de gestió

Aquesta fase dura tota la vida del projecte, ja que s'ha de gestionar des del mateix inici fins a la finalització del mateix.

Les tasques que es comprenen dins de la gestió del projecte són:

- Sol·licitud documentació DLA: Per a comunicar-se amb els diferents components domòtics de la casa, la interfície haurà de comunicar-se amb el servidor que enllaça amb cadascun d'ells, i el programador haurà d'usar aquesta documentació per familiaritzar-se i consultar-la durant la implementació del mateix.
- Reunions amb el client: Per obtenir els requisits del projecte, s'hauran de preparar varies reunions amb l'equip de treball del CVI, que complementen als del tutor del PFC. També s'inclouen reunions amb el tutor de projecte on s'explicaran els requisits ja definits que ha de tenir el software a desenvolupar.
- Reunions de seguiment: Generalment amb el tutor del projecte. Serveixen per aclarir dubtes, avaluar l'estat del projecte i prendre decisions en cas que hi hagi algun contratemps.
- Control del projecte: Dur a terme la pròpia gestió del projecte, com realitzar i revisar les estimacions, verificar els documents i codi a lliurar, gestionar els recursos disponibles, etc.

3.1.2. Tasques prèvies

Aquesta fase representa tota la feina que s'ha realitzat com a presa de contacte amb l'entorn i les necessitats del projecte, necessari per una major comprensió del problema a tractar i per minimitzar el temps requerit pel mateix a llarg termini.

Les activitats que queden incloses en aquesta fase són:

- Tecnologia implicada: Partint dels requisits imposats al propi projecte des de la Facultat d'Informàtica, s'han buscat les tecnologies que s'adaptaven millor a les característiques del projecte. Per trobar la tecnologia adient, s'han analitzat aquelles tecnologies més conegudes (considerades estàndards actuals en el món de la programació), així com altres tecnologies menys conegudes fruit de la recerca. Amb tot s'ha configurat un entorn de desenvolupament que incloïa una màquina amb Mac OS X 10.6, Java i XULrunner.

Les tasques per familiaritzar-se amb la tecnologia XUL (veure Secció 4.1.4, tecnologies candidates: XUL) constaran en entendre com funciona internament, quines similituds té amb altres tecnologies germanes com l'HTML i l'XML i quins són els seus punts forts i dèbils, juntament amb un estudi de les API disponibles.

- Presa de coneixements entorn (dificultats Cognitives / motrius): Per altra banda, per entendre millor quin és el problema existent i de quina forma s'està contribuint a solucionar-ho, s'ha cercat documentació sobre gent amb problemes cognitius i amb problemes motrius, per entendre millor algunes de les seves dificultats diàries i alhora comprendre millor algunes de les propostes i requisits que el projecte requereix.

3.1.3. Anàlisi i disseny

Aquesta fase es pot considerar la més important del projecte, ja que defineix la base principal del projecte: es dissenya la interfície, juntament amb una creació i valoració del cas d'ús del projecte, que seran usats en la següent fase d'implementació. Per tant, algun error en aquest punt seria de molt difícil esmena per la seva repercussió transversal a la resta de fases del projecte.

Les activitats incloses són:

- Reunions per definir els elements de la interfície gràfica: Amb els requisits obtinguts, es fa un estudi dels mateixos i s'avalua conjuntament quins elements es consideren rellevants en una interfície gràfica i quins no en quant a persones dependents.
- Creació del Storyboard: Es defineixen varies propostes d'interfície que contemplin els requisits detectats i els abordin de formes diferents. L'objectiu és discutir les diferents aproximacions i arribar a consens respecte la interfície final. Donada la naturalesa manual de la creació de l'Storyboard, no s'han inclòs en el document final; tot i que la interfície resultant es fruit d'aquests Storyboards.
- Refinament Storyboard: Amb l'input obtingut a mitjançant reunions amb l'equip d'experts del CVI, s'avaluen refinaments i ajustaments a les propostes inicials per obtenir un storyboard unificat amb les fortaleces de cadascuna de les propostes anteriors, juntament amb retocs proposats.
- Divisió de casos d'ús: Amb la proposta d'interfície definitiva acceptada, es consideren els casos d'ús i es distingeix i agrupen segons les funcionalitats que estan aportant cadascun d'aquests casos. S'estableix també un ordre de prioritats en quant a la implementació dels mateixos, premiant aquelles funcions considerades bàsiques per a qualsevol dels dos patrons d'estudi base: funcionalitats basades en sons per a persones amb deficiències visuals, i funcionalitats basades en escaneig per a persones amb dificultats motrius [veure Annex "Mètodes d'Interacció amb una GUI"].

3.1.4. Implementació

Un cop es disposa d'un anàlisi del problema inicial i s'han avaluat les diferents opcions, es tenen totes les peces necessàries per iniciar la implementació de la interfície. L'estructura a seguir en aquesta fase serà:

- Creació del directori del sistema: La tecnologia XUL requereix que es segueixi una estructura concreta quan es realitzen aplicacions usant aquest llenguatge. En la primera fase, es crearan les estructures necessàries per executar una aplicació XUL en entorns amb més d'una plataforma. Caldrà adaptar aquesta estructura a les peculiaritats dels sistemes Macintosh, Windows i Linux, tot i que aquesta tecnologia ja preveu aquestes peculiaritats i, per tant, està preparada per facilitar aquesta feina.
- Preparació de l'entorn: El següent pas en la implementació, tracta de la configuració de l'entorn d'execució, preparar l'aplicació per a que es comporti dins dels paràmetres desitjats.
- Implementació funcionalitats: Seguint les guies de programació d'una aplicació XUL, s'implementaran les funcionalitats del sistema seguint l'ordre de prioritats establert durant la fase d'especificació i disseny

3.1.5. Proves

Tal i com es van implementant funcionalitats, aquestes també s'han de provar per assegurar que el funcionament és l'esperat i així poder entregar un producte el més robust possible.

Hi ha dos grans blocs diferenciadors pel que fa a les proves del sistema. D'un cantó, tenim totes aquelles proves que es realitzen per comprovar els components de forma individual; per un altre cantó, un cop s'han comprovat de forma individual, s'ha de garantir que cadascuna de les funcionalitats interactua correctament amb la resta d'elements del programa.

Així doncs, aquesta fase queda definida amb les següents activitats:

- Definició del pla de proves: S'estructuren les funcionalitats implementades per seguir una lògica alhora de fer les proves, es jerarquitzen els elements a provar per optimitzar el temps.
- Proves unitàries: es mira cada element o conjunt d'elements de forma individual per assegurar el funcionament correcte i adient. S'inclou l'estudi de les pantalles,

botons i altres elements, tal i com queda recollit en el diagrama Gantt que es mostra més endavant.

- Proves integrals: El seu objectiu es comprovar el funcionament del programa de forma completa, verificant que els elements, ja comprovats de forma individual, interactuen com s'espera entre ells. Aquestes proves es realitzen tant a casa com a les oficines del client.

3.1.6. Finalització projecte

Bàsicament, les tasques que es duen a terme en aquesta fase són les pròpies de documentació necessàries per poder fer la presentació i donar per finalitzat el projecte; conjuntament amb un estudi econòmic del projecte, un cop finalitzades totes les fases necessàries per obtenir un producte i es poden calcular costos en funció dels recursos i el temps invertits. La documentació que s'inclou es la següent:

- Generació de la Memòria: es tracta d'aquest mateix document. Descriu el desenvolupament del projecte i les decisions que s'han anat prenent al llarg del mateix.
- Annexos: documentació que serveix com a reforç al document general. Inclou:
 - Manual d'instal·lació: Document que detalla els passos necessaris per instal·lar el software, així com l'estructura de la mateixa per si es considera necessari fer canvis en el codi o actualitzacions de les tecnologies base.
 - Mètodes d'interacció amb una interfície gràfica: S'exposen els diferents elements i mètodes existents quan es tracta d'interactuar amb una interfície gràfica. Els elements d'aquest document s'han considerat en les fases d'anàlisi i disseny per contemplar interfícies gràfiques i funcionalitats que s'adaptessin a aquells dispositius majoritaris en les persones amb alguna deficiència motriu o cognitiva.
 - Consideracions en el disseny d'una interfície gràfica: Recull de tècniques pràctiques i teòriques a considerar quan es realitzen dissenys gràfics per a ser usats per persones amb alguna dificultat cognitiva o motriu.

3.2. Recursos

En aquesta secció mostrarem inicialment els perfils que les tasques i activitats observades requereixen per dur a terme el projecte amb garantia d'èxit. Posteriorment s'analitzarà quins d'aquests rols els pot suplir l'encarregat del projecte, quins s'han d'externalitat i quins s'han de reubicar per falta de recursos.

3.2.1. Recursos requerits

Amb les activitats que requereix el projecte a la vista, queda clar que aquest projecte necessita un equip d'experts que siguin capaços d'assessorar en qüestions relatives a que pot i no pot fer algú que veu disminuïda la seva capacitat cognitiva o motriu.

Tanmateix, es necessita un analista que s'encarregui de realitzar les reunions periòdiques de seguiment i fer una gestió dels canvis, creï els Storyboard, analitzi requeriments..

Per a la part de la creació de la interfície i la connexió amb altres mòduls que interconnectin la interfície amb els dispositius domòtics, es necessita un programador de perfil mitjà, amb experiència amb el desenvolupament d'interfícies i a poder ser amb tecnologies web i tecnologies client-servidor.

Tots els mòduls que s'implementin s'hauran de provar, i l'encarregat de fer aquestes proves ha d'assegurar un cert aïllament del grup encarregat de la implementació, a forma de disminuir errors involuntaris o mostrar-se massa permissiu amb les proves realitzades.

Finalment, la part de documentació no requereix cap habilitat especial i la podria fer qualsevol persona, on generalment s'assigna aquesta feina a un becari (si es disposa d'algun), o del mateix cap de projectes.

3.2.2. Recursos disponibles

Malgrat les necessitats en quant a recursos detectada per realitzar el projecte, una situació ideal indicaria que per cada rol definit seria necessari mínim una persona. Això asseguraria un grau de dedicació a cada apartat suficient, i alhora escurçaria el temps destinat al projecte, tot augmentant la qualitat del mateix.

La realitat, però, indica que cal ajustar dràsticament aquestes necessitats als recursos disponibles. D'aquesta forma, tots els rols considerats interns els

realitzarà una mateixa persona, l'encarregat del PFC. Tot i això, no tots els recursos són limitats, ja que es disposa d'una infraestructura amb gran varietat de dispositius domòtics necessaris per dur a terme proves durant el projecte. Tanmateix, la necessitat d'experts en l'àmbit de la psicologia, posseïdors dels coneixements relatius a l'àmbit del projecte, són aportats pel centre CVI. Finalment, altres requisits del projecte, com personal d'enllaç entre el CVI, la FIB i l'estudiant encarregat del PFC serà dut a terme per treballadors del centre CVI posats per la facultat.

3.3. Temps

Aquest projecte està adscrit a les peculiaritats d'un finançament realitzat per la ONCE, el qual el fa subjecte a la revisió i tutelat de l'empresa sol·licitant del projecte. Així doncs, les restriccions temporals del projecte venen marcades per agents externs al projecte, tot i que no per això són menys rellevants. Ens trobem en una limitació temporal que situa la finalització del projecte a l'inici de l'any 2011.

Altres factors com la disponibilitats dels recursos necessaris per a dur a terme el projecte no s'han tingut en consideració ja que es considera que aquests estaran sempre disponibles per a la durada del projecte, i en cas que aquest s'allargués de forma extraordinària s'haurien de reavaluar les conseqüències del contratemps també en clau de disponibilitat de recursos i temps de projecte.

4. Anàlisi inicial

Realitzar el projecte amb garanties requereix que prèviament s'hagin analitzat tantes alternatives com sigui possible, de manera que es puguin prendre decisions informades. Aquestes alternatives són d'una vessant tècnica i varien temporalment en funció de la tecnologia disponible.

Les decisions que s'han pres han considerat requisits inherents del projecte tals com la portabilitat, facilitat per modificar el codi font i integració del producte final amb el Look & Feel del Sistema Operatiu.

4.1. Tecnologies candidates

Per a dur a terme un tasca com la proposada, es necessita escollir bé la tecnologia a emprar en el projecte, doncs cada tecnologia té els seus punts forts i febles i escollir-ne una errònia comporta problemes i dificultats afegides que podrien arribar a ser insalvables.

Per aquest motiu, s'han mirat un conjunt de tecnologies candidates de les que s'han derivat les conclusions següents:

4.1.1. Java

Té l'avantatge de ser una tecnologia de desenvolupament multi plataforma, robusta i provada en sectors empresarials; tot i que és conegut que les interfícies no són tant senzilles de tractar com en altres llenguatges i tampoc s'adapten a l'aparença del S.O. en el que s'executen. A més a més l'avantatge principal que suposa crear un únic codi precompilat que s'executa en qualsevol entorn d'escriptori (Linux, Windows, Mac) també suposa que el producte final es poc modificable (es necessitarien les fonts i conèixer els mètodes de compilació necessaris).

4.1.2. C++

Aquests tipus de tecnologies, tot i que son molt vàlides per entorns d'una sola plataforma, representen l'inconvenient de que han de ser compilades de forma independent per a cada tipus de S.O. al que van destinats, i per tant no només els executables no son multi plataforma sinó que molt probablement el codi font necessiti ajustaments propis de cada plataforma per fer-lo funcionar. A més a més no hi un estàndard pel que es refereix a la interfície, cosa que faria necessari buscar alguna API per a aquesta tasca (les llibreries QT en serien un exemple).

4.1.3. Tecnologies Web

També es van tenir en consideració tecnologies més comuns avui dia com ho és el paquet HTML, CSS, Javascript pel que fa a la presentació i PHP pel que fa al model de dades. Tot i que a priori semblava una solució que s'adapta amb facilitat als requisits del projecte (permet crear interfícies a partir d'XML que després son altament modificables), presentava el greu inconvenient que necessita configurar algun tipus de servidor per a funcionar, fet que limita molt la possible implantació en el client final, ja que, d'usar-se aquest sistema, l'aplicació final hauria d'executar dos servidors per funcionar: el web (per oferir una lògica a la interfície) i el DLA (per comunicar-se amb els elements domòtics de la casa).

4.1.4. XUL

Aquesta tecnologia té el seu origen en les sigles *XML-based User Interface*, i es tracta d'un derivat de les tecnologies Web desenvolupat per Mozilla que destaca per oferir una definició d'interfícies gràfiques simple i portable, sense requerir necessàriament una tecnologia de servidor per al correcte funcionament. De fet es tracta de la tecnologia que hi ha al darrere de les interfícies gràfiques de les aplicacions més famoses d'aquest grup: Firefox (explorador Web) i Thunderbird (gestor de correus).

Tot i no ser un estàndard, aquesta tecnologia va estar pensada per ser-ho des de l'inici, i incorpora tot un conjunt d'eines per a la manipulació d'interfícies gràfiques que abasten la creació de menús, panells i barres d'eines, entre d'altres, que la fan una solució molt més potent i ràpida que altres com l'HTML. Habitualment les interfícies XUL utilitzen Javascript per a definir la seva lògica, mentre que també es poden recolzar en altres tecnologies web com el CSS per a definir els estils visuals. Tot i això, XUL pot comunicar-se amb pràcticament qualsevol llenguatge clàssic, com el Java o el C++.

Finalment cal destacar que la majoria d'aplicacions multi plataforma que usen altres llenguatges s'han de desenvolupar usant característiques pròpies d'una plataforma específica, fent que el desenvolupament del software sigui costós i consumeixi més temps; per aquesta raó es van desenvolupar solucions multi plataforma com Java, però com avantatja del XUL a aquesta tecnologia, una interfície gràfica desenvolupada amb XUL pot ser implementada i modificada significativament més ràpid i fàcilment; a més dels avantatges que suposa per a requisits com el de manteniment poder accedir al codi font a partir de l'executable.

Exposades les característiques de cadascuna de les tecnologies candidates, s'ha optat per usar XUL com a tecnologia base, doncs es tracta d'una tecnologia oberta, pensada per ser multi plataforma i que es pot usar tant com aplicació independent d'escriptori com aplicació Web; i el fet que estigui basat en tecnologies web permet una alta flexibilitat en la definició de la interfície gràfica, així com possibilita la posterior modificació de la mateixa amb gran facilitat.

4.1.5. Reproducció de so

Pel que fa a l'audició, s'han mirat les diferents opcions per a reproduir so basades amb la lectura directe de text en pantalla; conclouent que la immaduresa que presenten aquestes tecnologies, tant a nivell de multi plataforma (les solucions basades en Windows no son compatibles ni amb Linux ni amb Mac i viceversa) com a nivell de resultats esperats (en general les veus son robòtiques i no s'adapten ni a l'entonació ni a l'accent desitjats) la fan una solució inviable. Així doncs, el projecte s'ha decantat per la via de reproduir sons associats a objectes, on clarament apareixen inconvenients tals com la mida que ocupa l'aplicació final, així com la necessitat d'enregistrar tot allò que es vulgui reproduir; tot i això, amb aquesta solució es mantenen intactes les opcions d'independència del Sistema Operatiu, així com es millora el nivell de personalització auditiu, fent-lo més reconeixible per a les persones a qui va destinat el projecte.

4.2. Tecnologies implicades

Finalitzat el període d'anàlisi de tecnologies, i un cop escollides quines serien les principals, el projecte ha requerit compaginar varies d'aquestes tecnologies per dur a terme cadascuna de les tasques encomanades. A continuació es mostra cadascuna de les àrees en les que s'han aplicat les tecnologies implicades:

- Estàndards web: CSS, Javascript i XML. La tecnologia CSS es àmpliament usada per a definir comportaments visuals (posicionament de contingut, aparença, etc); Javascript és emprat en el món de les tecnologies Web per a presentar contingut calculat en la banda del client, proporciona un llenguatge flexible i altament adaptable que en el projecte aporta la capacitat necessària a l'aplicació per a gestionar de forma interna totes les operacions de càlcul de la interfície; finalment l'XML és emprat per a transmetre informació, ja sigui des de l'aplicació d'administració cap a la d'usuari, com per a guardar i carregar configuracions dins de la mateixa aplicació.

- Basats en estàndards: XUL. Aquesta tecnologia proporciona al sistema l'estructura bàsica de la interfície. Defineix els diferents elements a visualitzar i les relacions entre ells, permetent que el contingut es gestioni de forma dinàmica mitjançant altres tecnologies de manipulació de dades. Alhora també aporta de forma nativa les capacitats de reproducció d'imatge i so (i vídeo si fos necessari) que el sistema requereix.
- Altres tecnologies: Java (classes, connexió API) i C++ (executable Windows i Linux). Aquestes dues tecnologies no es poden considerar indispensables en quant a l'elaboració de la interfície, però factors externs a aquesta han fet necessari que s'incorporin aquestes tecnologies al projecte. Concretament, Java es necessari per comunicar la interfície amb la API que controla els diferents dispositius domòtics de la casa; per altra banda, es necessitava crear un executable per entorns Linux i Windows que evités que el programa necessités d'una línia de comandes per ser cridat, i això es va solucionar creant mini executables nadius per a cadascuna d'ambdues plataformes (els S.O. Macintosh no necessiten de cap executable per executar-se de forma nativa).

4.3. Proves tecnològiques

Amb la tecnologia decidida, s'han realitzat un seguit de proves amb l'objectiu d'establir una base de coneixement sobre la tecnologia amb la que després realitzar el projecte; així com també dur a terme proves sobre característiques específiques que requereix el projecte i que, tot i que es té el coneixement de que és factible realitzar-les, no s'ha comprovat encara la seva senzillesa o dificultat.

Així doncs les proves es van dividir en tres fases:

4.3.1. Recaptació documentació XUL

Es tractava de comprovar quines funcionalitats s'especificaven a l'API, prendre contacte amb pàgines d'on extreure recursos per a aquesta tecnologia i observar grups de discussió per assabentar-se de les dificultats típiques que altres programadors trobaven.

4.3.2. Proves unitàries

Com si es tractés del test d'una aplicació, es va començar provant parts de l'API per separat. Inicialment es van fer estructures que simulaven finestres d'aplicacions, amb components com series pestanyes, botons i altres estructures bàsiques a qualsevol interfície.

Tot seguit es va passar a provar altres elements més crítics del projecte, com la reproducció de sons, imatges i vídeos, coneixent les virtuts i límits que aportava aquesta tecnologia.

Finalment, es van acabar les proves unitàries amb tests d'interconnectivitat amb la tecnologia Java, que suposava la tecnologia de rerefons de la interfície.

4.3.3. Proves integrals

Provada cada tecnologia per separat, es necessitava provar la interacció entre tecnologies i elements de la interfície. D'aquesta manera, es va crear una aplicació reduïda destinada a provar les tres tecnologies principals que haurien d'interactuar més endavant en l'aplicació final:

- Mètodes per canviar l'aparença de la interfície 'al vol'. (CSS/XUL – Javascript)
- Interacció entre Interfície i Java i viceversa (XUL – Javascript – Java).
- Presentació dinàmica de continguts audiovisuals (interacció XUL amb Javascript)

5. Especificació

Aquesta etapa del projecte duu a terme una descripció detallada sobre el comportament extern del sistema; es a dir, determina el funcionament des del punt de vista de l'usuari final. El posterior disseny i implementació del sistema estaran basats en l'especificació, de forma que es important prendre bones decisions en aquesta fase del projecte.

Per generar la documentació, es realitzarà primerament el model de casos d'ús del sistema, juntament amb els actors implicats. També s'especificaran tots els contractes de les operacions del sistema. S'acaba aquesta secció amb el model conceptual, per facilitar la identificació dels objectes que formen part del sistema i les seves relacions.

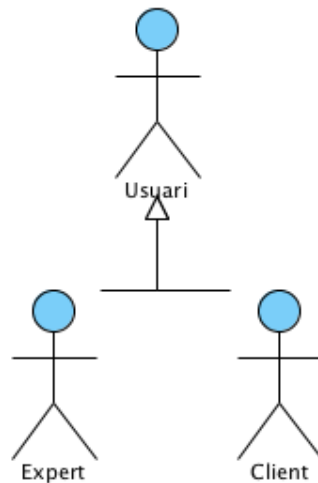
5.1. Model de casos d'ús

Els casos d'ús són una forma de representar la interacció que tindrà un usuari (ja sigui persona o màquina) amb el sistema. El que es persegueix amb això es allunyar-se de l'abstracció tècnica que acompanya tot desenvolupament de software i centrar-nos en la interacció de l'usuari amb el sistema; finalitat del qual és realitzar amb èxit un requeriment funcional (veure Secció 2, apartat 3: Requisits).

5.1.1. Actors del sistema

Els actors són entitats del sistema que participen en els casos d'ús. Per tant, cal identificar-los i relacionar-los en cadascuna de les tasques que realitzen de forma activa.

Els actors que es tindran en consideració per especificar els casos d'ús son els següents:



Il·lustració 1. Diagrama d'actors

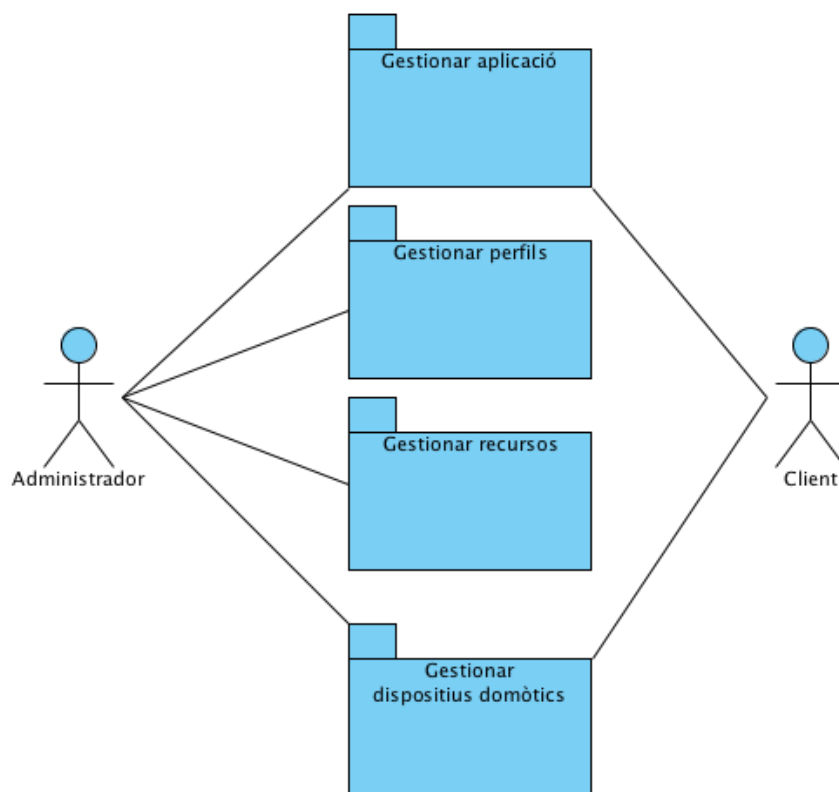
D'una banda tenim els usuaris del tipus 'Administrador', que faran servei de l'aplicació per configurar diferents paràmetres i adaptar la interfície a les necessitats de l'usuari 'Client', així com també definiran com es volen disposar en la interfície els dispositius domòtics. Per altra banda, els usuaris 'Client' faran ús de la interfície gràfica per gestionar els diferents elements domòtics de la llar.

5.2. Diagrama dels casos d'ús

Els diagrames de cas d'ús són un tipus de diagrama en UML que serveix per modelar aspectes dinàmics del sistema. Són importants per a modelar el comportament d'un sistema, subsistema o classe.

Els casos d'ús representen una funcionalitat o conjunt de funcionalitats del sistema; cadascun d'aquests paquets mostra un conjunt de casos d'ús, actors i les seves relacions. En el projecte s'ha realitzat una agrupació per paquets segons el tipus de funcionalitats que aporten.

Els diferents paquets de casos d'ús del sistema són els següents:



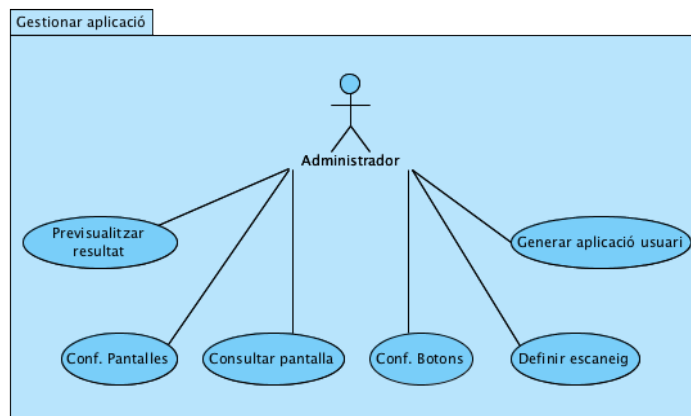
Il·lustració 2. Classificació dels casos d'ús

Cadascun d'aquests casos està agrupat en funció de les funcions que tenen en comú:

- Gestionar aplicació: agrupa totes les funcionalitats que serveixen per adaptar la interfície gràfica a les necessitats dels usuaris finals. Es tracta de casos d'ús que fan referència a la mida i característiques d'un boto, nombre de pantalles, colors del sistema, etc.
- Gestionar perfils: Son totes aquelles funcionalitats que tracten amb l'estat del sistema des de la perspectiva de la persistència de dades.
- Gestionar recursos: De manera similar a la gestió de perfils, la gestió de recursos treballa amb tots els casos d'ús que es relacionen amb la manipulació d'imatges i sons; també es persistència de dades, però no afecten directament a l'estat del sistema.
- Gestió Dispositius Domòtics: Es refereix a la comunicació de la interfície gràfica amb el sistema DLA per controlar els diferents dispositius existents a la llar.

5.3. Especificació dels casos d'ús

5.3.1. Gestionar aplicació



Il·lustració 3. Casos d'ús 'Gestionar aplicació'

Cas d'ús	# 1.01. Configurar pantalles
Àmbit	Gestió CVI
Actor primari	Usuari administrador del sistema
Stakeholders i interessats	L'actor primari
Precondicions	-
Garanties d'èxit	El sistema té les dades de les pantalles emmagatzemades
Activació	L'actor primari vol modificar les dades de la interfície
Curs típic d'esdeveniments	<ol style="list-style-type: none"> 1. L'actor primari indica al sistema que vol modificar les dades de l'aplicació. 2. El sistema mostra en la zona de configuracions totes les opcions que es poden editar. 3. L'actor primari modifica aquells atributs que considera necessaris. 4. El sistema mostra els canvis realitzats a la pantalla en ús. 5. Acaba el cas d'ús.

Taula 1. Cas d'ús "Configurar pantalles"

Cas d'ús	# 1.02. Consultar pantalla
Àmbit	Gestió CVI
Actor primari	Usuari administrador del sistema
Stakeholders i interessats	L'actor primari
Precondicions	La pantalla es carrega en el sistema
Garanties d'èxit	El sistema té les dades de les pantalles emmagatzemades
Activació	L'actor primari vol consultar les dades d'una pantalla d'usuari.
Curs típic d'esdeveniments	<ol style="list-style-type: none"> 1. El sistema mostra en un llistat totes les pantalles del sistema. 2. L'actor primari selecciona la pantalla corresponent. 3. El sistema mostra les dades de la pantalla seleccionada. 4. Acaba el cas d'ús.

Taula 2. Cas d'ús "Consultar pantalla"

Cas d'ús	# 1.03. Configurar botons
Àmbit	Gestió CVI
Actor primari	Usuari administrador del sistema
Stakeholders i interessats	L'actor primari
Precondicions	El botó a configurar existeix
Garanties d'èxit	El sistema té les dades del botó emmagatzemades
Activació	L'actor primari vol modificar les dades d'un botó
Curs típic d'esdeveniments	<ol style="list-style-type: none"> 1. L'actor primari indica al sistema que vol modificar les dades d'un botó. 2. El sistema mostra un requadre amb totes les opcions que es poden editar: text, imatge, sons i acció al prémer-lo. 3. L'actor primari modifica aquells atributs que considera necessaris. 4. El sistema mostra els canvis realitzats al botó seleccionat. 5. Acaba el cas d'ús.
Extensions	<p>3a. L'actor primari cancel·la l'operació.</p> <p>3a1. Acaba el cas d'ús.</p>

Taula 3. Cas d'ús "Configurar botons"

Cas d'ús	# 1.04. Definir escaneig
Àmbit	Gestió CVI
Actor primari	Usuari administrador del sistema
Stakeholders i interessats	L'actor primari
Precondicions	-
Garanties d'èxit	El sistema té les dades d'escaneig emmagatzemades
Activació	L'actor primari vol modificar el comportament de les opcions d'escaneig de botons.
Curs típic d'esdeveniments	<ol style="list-style-type: none"> 1. L'actor primari indica al sistema que vol modificar el comportament de les opcions d'escaneig de botons. 2. El sistema mostra en la zona de configuracions totes les opcions que es poden editar. 3. L'actor indica al sistema que desitja activar l'escaneig. 4. El sistema habilita en mode escriptura les opcions associades al escaneig. 5. L'actor primari modifica aquells atributs que considera necessaris. 6. Acaba el cas d'ús
Extensions	<p>3a. L'actor primari indica al sistema que desitja desactivar l'escaneig.</p> <p>3a1. El sistema inhabilita el mode escriptura en les opcions associades a l'escaneig.</p> <p>3a2. Acaba el cas d'ús.</p>

Taula 4. "Definir escaneig"

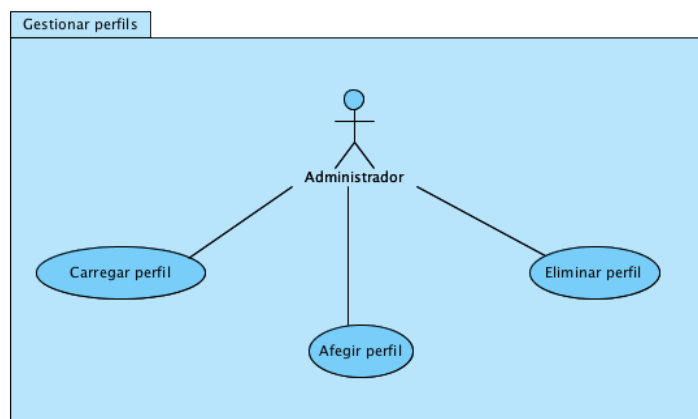
Cas d'ús	# 1.05. Vista prèvia del resultat
Àmbit	Gestió CVI
Actor primari	Usuari administrador del sistema
Stakeholders i interessats	L'actor primari
Precondicions	-
Garanties d'èxit	El sistema genera la pantalla del client final amb les configuracions d'aplicació actuals.
Activació	L'actor primari vol visualitzar la configuració de l'aplicació tal i com la percebrà el l'usuari destinatari.
Curs típic d'esdeveniments	<ol style="list-style-type: none"> 1. L'actor primari indica al sistema que vol visualitzar la configuració actual tal i com la percebrà l'usuari final. 2. El sistema mostra en una nova finestra la configuració de l'aplicació adaptada a les característiques pròpies de la versió per al client final. 3. Acaba el cas d'ús

Taula 5. Cas d'ús "Vista prèvia del resultat"

Cas d'ús	# 1.06. Generar aplicació d'usuari
Àmbit	Gestió CVI
Actor primari	Usuari administrador del sistema
Stakeholders i interessats	L'actor primari
Precondicions	-
Garanties d'èxit	El sistema genera una aplicació auto continguda amb la informació de l'estat del sistema actual.
Activació	L'actor primari vol crear l'aplicació que usará l'usuari destinatari.
Curs típic d'esdeveniments	<ol style="list-style-type: none"> 1. L'actor primari indica al sistema que vol generar l'aplicació per a l'usuari final. 2. El sistema mostra un llistat amb els diferents Sistemes Operatius (S.O.) disponibles. 3. L'actor primari marca aquells S.O. per als quals vol generar l'aplicació. 4. El sistema genera per a cada S.O. marcat una carpeta amb els executables i les configuracions de l'aplicació d'usuari en base a l'estat actual del sistema. 5. Acaba el cas d'ús

Taula 6. Cas d'ús "Generar aplicació d'usuari"

5.3.2. Gestionar perfils



Il·lustració 4. Casos d'ús 'Gestionar perfils'

Cas d'ús	# 2.01. Carregar perfil
Àmbit	Gestió CVI
Actor primari	Usuari administrador del sistema
Stakeholders i interessats	L'actor primari
Precondicions	El perfil existeix prèviament.
Garanties d'èxit	El sistema té les dades del perfils carregades
Activació	L'actor primari vol visualitzar una configuració de pantalla guardada pel sistema.
Curs típic d'esdeveniments	<ol style="list-style-type: none"> 1. L'actor primari indica al sistema que vol carregar una configuració de pantalla prèviament guardada. 2. El sistema mostra un requadre amb tots els perfils emmagatzemats en el sistema. 3. L'actor primari selecciona el perfil corresponent i selecciona l'opció 'carregar'. 4. El sistema carrega les dades del perfil seleccionat. 5. Acaba el cas d'ús

Taula 7. Cas d'ús "Carregar perfil"

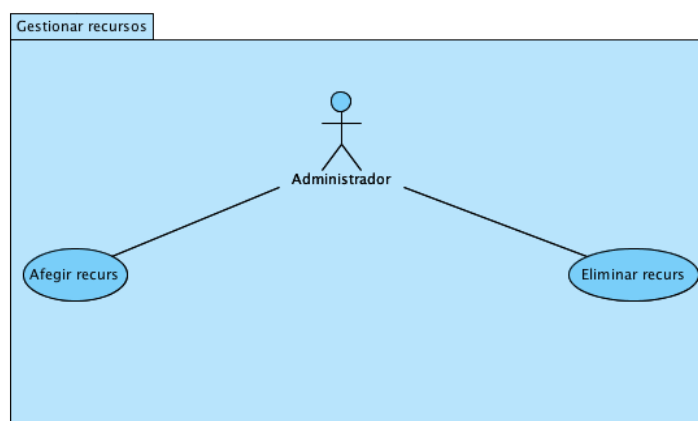
Cas d'ús	# 2.02. Afegir perfil
Àmbit	Gestió CVI
Actor primari	Usuari administrador del sistema
Stakeholders i interessats	L'actor primari
Precondicions	-
Garanties d'èxit	El sistema té les dades del perfils emmagatzemades
Activació	L'actor primari vol guardar l'estat de l'aplicació.
Curs típic d'esdeveniments	<ol style="list-style-type: none"> 1. L'actor primari indica al sistema que vol visualitzar els perfils. 2. El sistema mostra un requadre amb tots els perfils emmagatzemats en el sistema. 3. L'actor indica al sistema que vol guardar l'estat actual. 4. El sistema mostra un requadre per assignar nom al perfil. 5. L'actor primari introdueix un nom del perfil i indica al sistema que vol guardar els canvis. 6. El sistema genera el perfil actual amb el nom indicat. 7. Acaba el cas d'ús
Extensions	<p>3a. L'actor primari cancel·la l'operació. 3a1. Acaba el cas d'ús.</p> <p>6a. El nom indicat ja existeix en el sistema. 6a1. El sistema sobreescriu el fitxer ja existent amb l'estat actual de l'aplicació. 6a2. Acaba el cas d'ús.</p>

Taula 8. Cas d'ús "Afegir perfil"

Cas d'ús	# 2.03. Eliminar perfil
Àmbit	Gestió CVI
Actor primari	Usuari administrador del sistema
Stakeholders i interessats	L'actor primari
Precondicions	El perfil està emmagatzemat en el sistema
Garanties d'èxit	El sistema esborra les dades del perfils emmagatzemades
Activació	L'actor primari vol eliminar un estat d'aplicació emmagatzemat.
Curs típic d'esdeveniments	<ol style="list-style-type: none"> 1. L'actor primari indica al sistema que vol visualitzar els perfils. 2. El sistema mostra un requadre amb tots els perfils emmagatzemats en el sistema. 3. L'actor primari selecciona el perfil que vol eliminar i prem la opció d'esborrat. 4. El sistema elimina el perfil emmagatzemat. 5. Acaba el cas d'ús
Extensions	<p>3a. L'actor primari cancel·la l'operació.</p> <p>3a1. Acaba el cas d'ús.</p>

Taula 9. Cas d'ús "Eliminar perfil"

5.3.3. Gestionar recursos



Il·lustració 5. Casos d'ús 'Gestionar recursos'

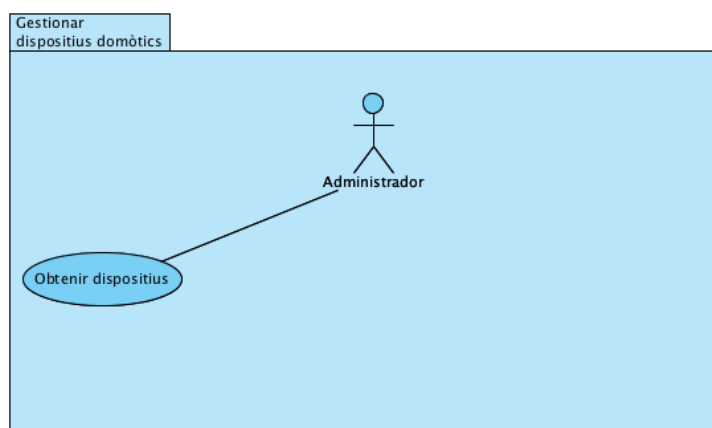
Cas d'ús	# 3.01. Afegir recurs
Àmbit	Gestió CVI
Actor primari	Usuari administrador del sistema
Stakeholders i interessats	L'actor primari
Precondicions	-
Garanties d'èxit	El sistema emmagatzema les dades del recurs sol·licitat
Activació	L'actor primari vol emmagatzemar un recurs visual o auditiu al sistema.
Curs típic d'esdeveniments	<ol style="list-style-type: none"> 1. L'actor primari indica al sistema que vol visualitzar els recursos de tipus visuals o auditius. 2. El sistema mostra un requadre amb tots els recursos emmagatzemats en el sistema del tipus indicat. 3. L'actor primari indica que vol afegir un recurs. 4. El sistema mostra un diàleg per escollir un fitxer del tipus desitjat del Sistema Operatiu. 5. L'usuari escull el fitxer i indica al sistema que vol guardar els canvis. 6. El sistema emmagatzema el fitxer. 7. Acaba el cas d'ús.
Extensions	<p>3a. L'actor primari cancel·la l'operació. 3a1. Acaba el cas d'ús.</p> <p>6a. El recurs ja existeix en el sistema 6a1. El sistema mostra un missatge d'error i avorta l'execució. 6a2. Acaba el cas d'ús.</p>

Taula 10. Cas d'ús "Afegir recurs"

Cas d'ús	# 3.02. Eliminar recurs
Àmbit	Gestió CVI
Actor primari	Usuari administrador del sistema
Stakeholders i interessats	L'actor primari
Precondicions	El recurs està emmagatzemant en el sistema
Garanties d'èxit	El sistema esborra les dades del recurs emmagatzemades
Activació	L'actor primari vol eliminar un recurs visual o auditiu del sistema.
Curs típic d'esdeveniments	<ol style="list-style-type: none"> 1. L'actor primari indica al sistema que vol visualitzar els recursos de tipus visuals o auditius. 2. El sistema mostra un requadre amb tots els recursos emmagatzemats en el sistema del tipus indicat. 3. L'actor primari selecciona un recurs del llistat i indica al sistema que vol eliminar-lo. 4. El sistema esborra el recurs. 5. Acaba el cas d'ús.
Extensions	<p>3a. L'actor primari cancel·la l'operació.</p> <p>3a1. Acaba el cas d'ús.</p>

Taula 11. Cas d'ús "Eliminar recurs"

5.3.4. Gestionar Dispositius Domòtics

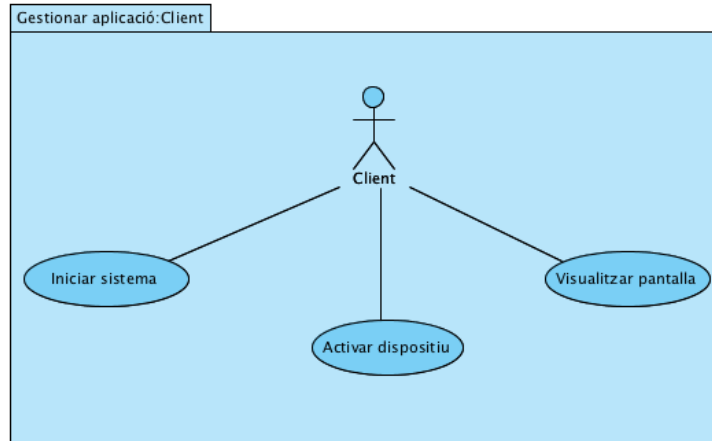


Il·lustració 6. Casos d'ús 'Obtenir dispositius'

Cas d'ús	# 4.01. Obtenir dispositius
Àmbit	Gestió CVI
Actor primari	Usuari administrador del sistema
Stakeholders i interessats	L'actor primari
Precondicions	El servidor DLA està activat i és visible pel sistema
Garanties d'èxit	El sistema enregistra els dispositius que hi ha a la casa
Activació	L'actor primari vol consultar els dispositius domòtics existents en la casa.
Curs típic d'esdeveniments	<ol style="list-style-type: none"> 1. L'actor primari indica al sistema que vol consultar els dispositius domòtics. 2. El sistema es connecta al DLA i mostra en un llistat els dispositius disponibles. 3. L'actor primari selecciona el dispositiu domòtic desitjat. 4. El sistema retorna el un llistat amb el conjunt d'accions i localitzacions que s'apliquen al dispositiu seleccionat. 5. Acaba el cas d'ús.
Extensions	<p>3a. L'actor primari cancel·la l'operació.</p> <p>3a1. Acaba el cas d'ús.</p>

Taula 12. Cas d'ús "Obtenir dispositius"

5.3.5. Gestionar aplicació (client)



Il·lustració 7. Casos d'ús "Gestionar aplicació (client)"

Cas d'ús	# 5.01. Iniciar Sistema
Àmbit	Gestió CVI
Actor primari	Usuari client del sistema
Stakeholders i interessats	L'actor primari
Precondicions	El perfil existeix prèviament.
Garanties d'èxit	El sistema té les dades del perfils carregades
Activació	L'actor primari vol iniciar el sistema.
Curs típic d'esdeveniments	<ol style="list-style-type: none"> 1. L'actor primari inicia el sistema. 2. El sistema cerca el perfil per defecte i carrega les dades. 3. Acaba el cas d'ús

Taula 13. Cas d'ús "Iniciar Sistema"

Cas d'ús	# 5.02. Visualitzar pantalla
Àmbit	Gestió CVI
Actor primari	Usuari client del sistema
Stakeholders i interessats	L'actor primari
Precondicions	La pantalla es carrega en el sistema
Garanties d'èxit	El sistema té les dades de les pantalles emmagatzemades
Activació	L'actor primari vol consultar les dades d'una pantalla d'usuari.
Curs típic d'esdeveniments	<ol style="list-style-type: none"> 1. L'actor primari executa l'acció d'un botó configurat per carregar una pantalla. 2. El sistema mostra les dades de la pantalla seleccionada. 3. Acaba el cas d'ús.

Taula 14. Cas d'ús "Visualitzar pantalla"

Cas d'ús	# 5.03. Activar dispositius
Àmbit	Gestió CVI
Actor primari	Usuari client del sistema
Stakeholders i interessats	L'actor primari
Precondicions	El servidor DLA està activat i és visible pel sistema
Garanties d'èxit	El sistema comunica al DLA els dispositius que s'ha d'accionar.
Activació	L'actor primari vol accionar un dispositius domòtic del sistema.
Curs típic d'esdeveniments	<ol style="list-style-type: none"> 1. L'actor primari indica al sistema que vol accionar un dispositiu domòtic. 2. El sistema es connecta al DLA. 3. El sistema envia al DLA la petició amb l'identificador del dispositiu i l'acció a realitzar. 4. Acaba el cas d'ús.
Extensions	<ol style="list-style-type: none"> 3a. El sistema no es pot comunicar amb el dispositiu indicat. <ol style="list-style-type: none"> 3a1. El sistema mostra un missatge d'error pertinent (DLA no trobat, dispositiu inexistent ...). 3a2. Acaba el cas d'ús.

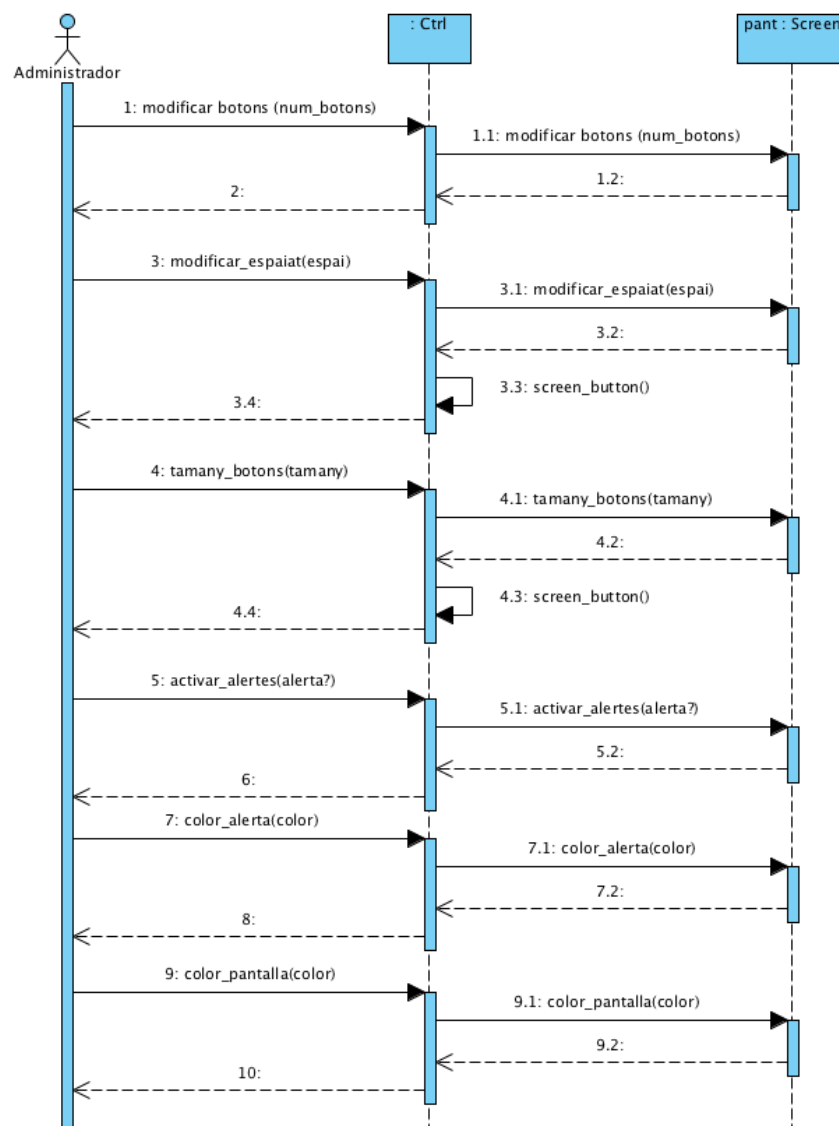
Taula 15. Cas d'us "Activar dispositius"

5.4. Model de comportament

En aquest apartat es detallarà els diferents casos d'ús explicats anteriorment, així com també els contractes de les operacions que els hi acompanyen.

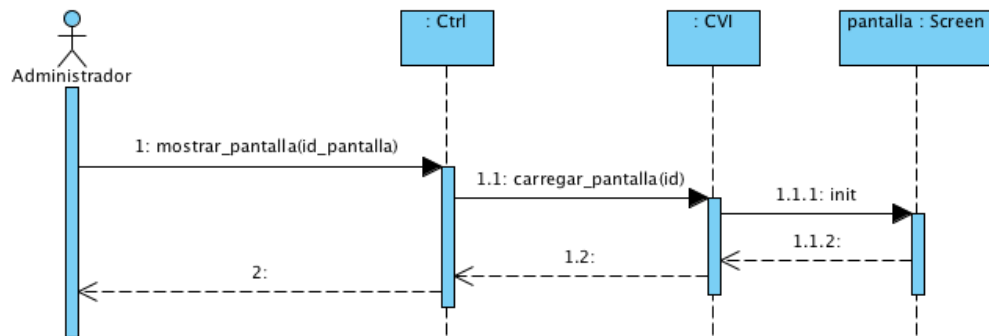
5.4.1. Gestionar aplicació

5.4.1.01. Configurar pantalles



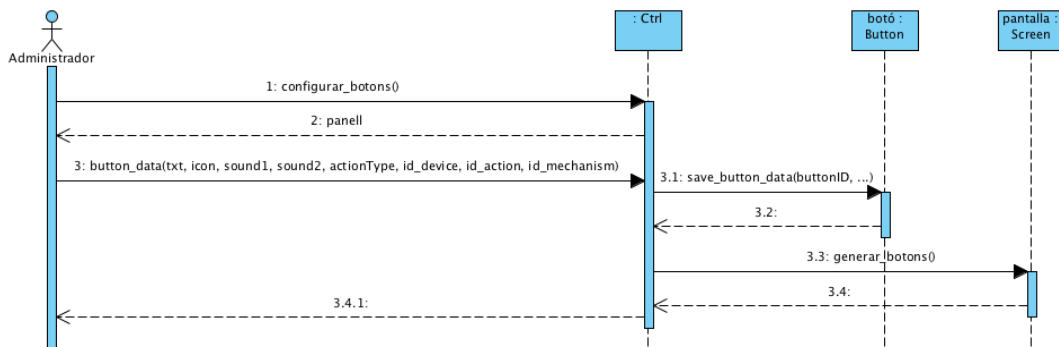
Il·lustració 8. Cas d'ús configurar pantalles

5.4.1.02. Consultar pantalla



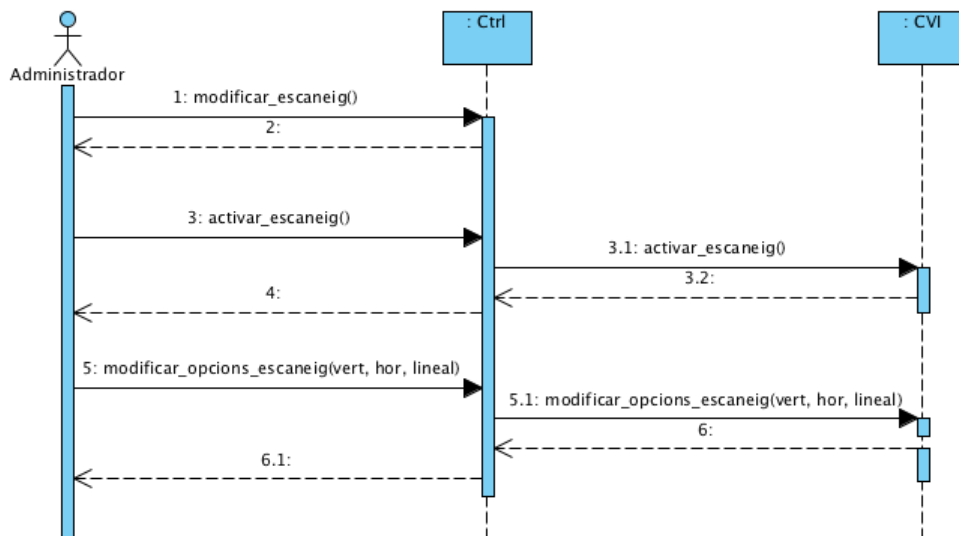
Il·lustració 9. Cas d'ús consultar pantalla

5.4.1.03. Configurar botons



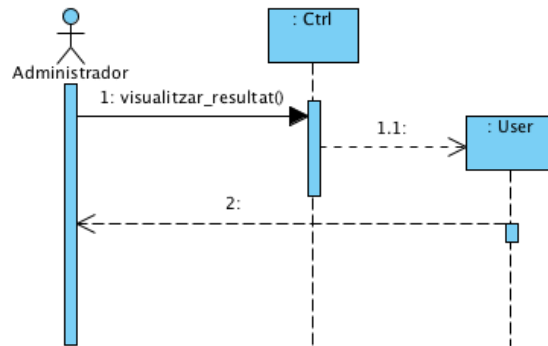
Il·lustració 10. cas d'ús configurar botons

5.4.1.04. Definir escaneig



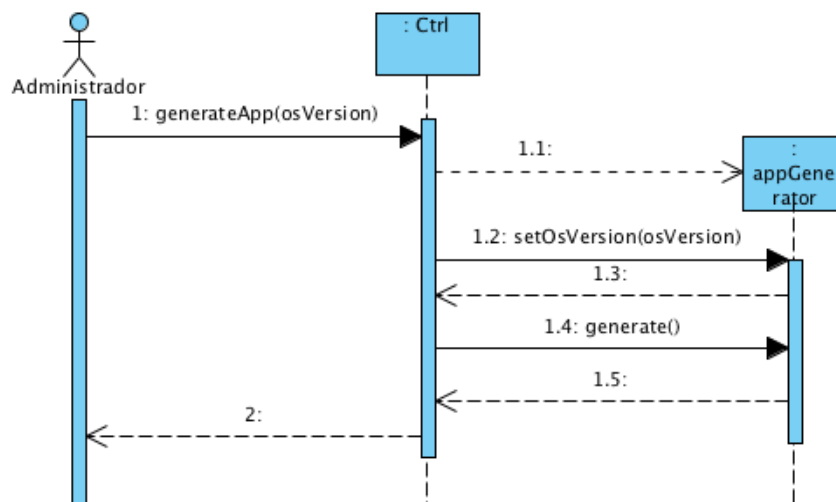
Il·lustració 11. Cas d'ús definir escaneig

5.4.1.05. Vista prèvia del resultat



Il·lustració 12. Cas d'ús vista prèvia del resultat

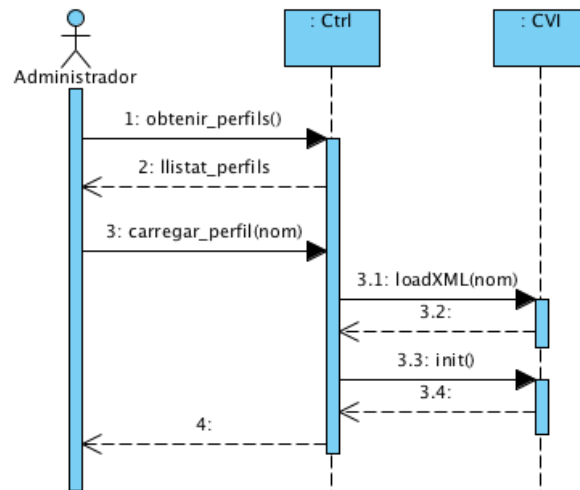
5.4.1.06. Generar aplicació usuari



Il·lustració 13. Cas d'ús Generar aplicació d'usuari

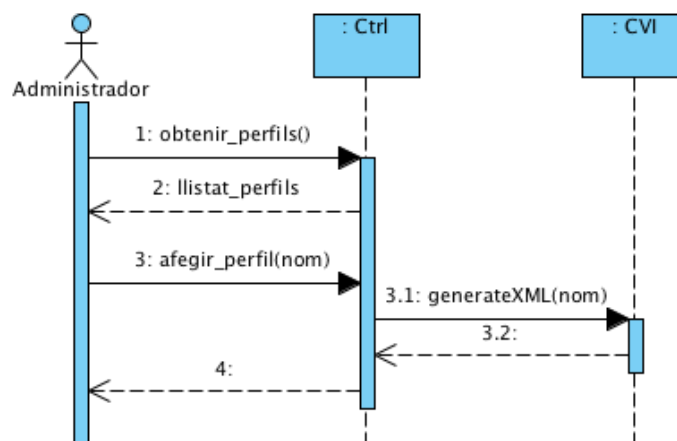
5.4.2. Gestionar perfils

5.4.2.01. Carregar perfil



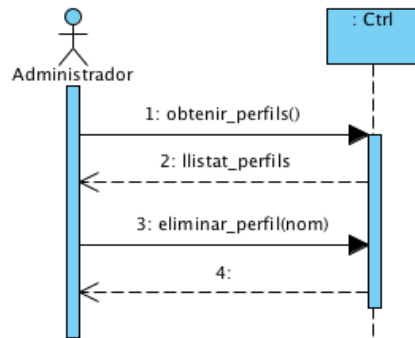
Il·lustració 14. Cas d'ús Carregar perfil

5.4.2.02. Afegir perfil



Il·lustració 15. Cas d'ús Afegir perfil

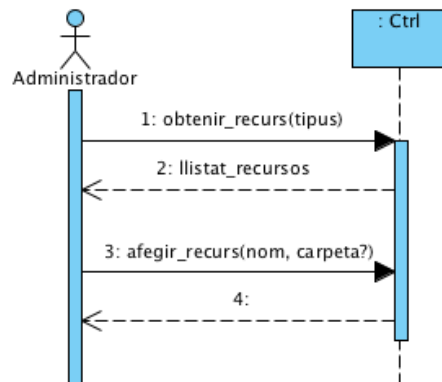
5.4.2.03. Eliminar perfil



Il·lustració 16. Cas d'ús Eliminar perfil

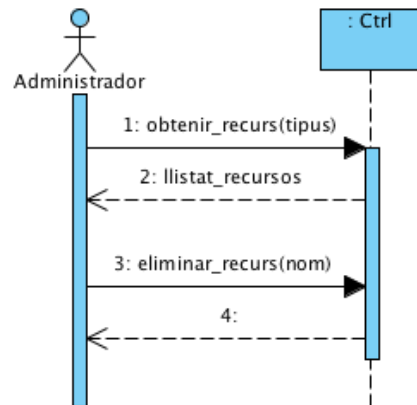
5.4.3. Gestionar recursos

5.4.3.01. Afegir recurs



Il·lustració 17. Cas d'ús Afegir recurs

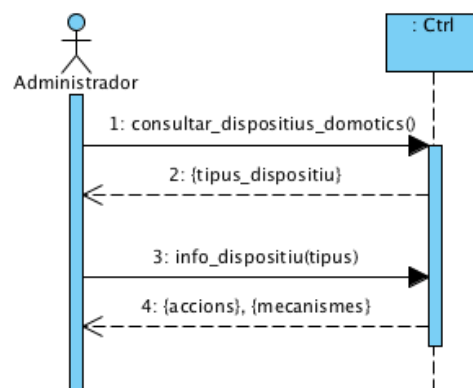
5.4.3.02. Eliminar recurs



Il·lustració 18. Cas d'ús Eliminar recurs

5.4.4. Gestionar DLA

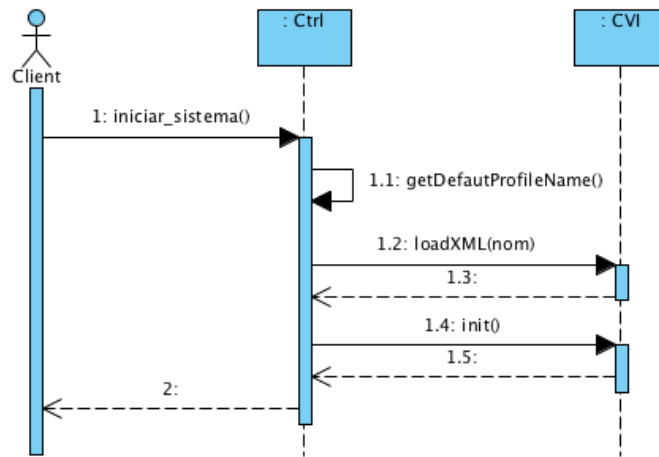
5.4.4.01. Obtenir dispositius



Il·lustració 19. Cas d'ús Obtenir dispositius

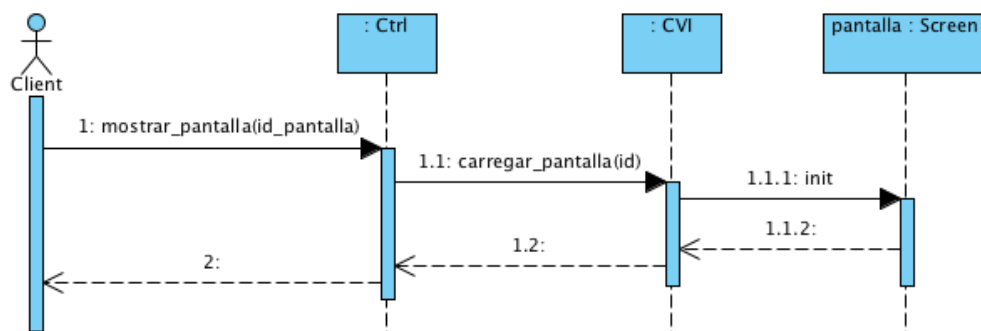
5.4.5. Gestionar aplicació (client)

5.4.5.01. Carregar perfil



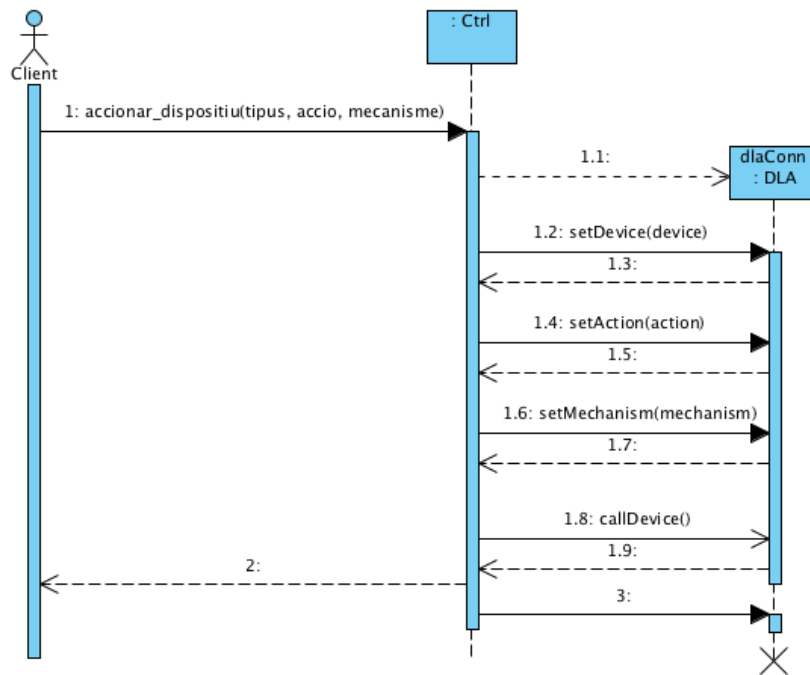
Il·lustració 20. Cas d'ús Carregar perfil

5.4.5.02. Visualitzar pantalla



Il·lustració 21. Cas d'ús Visualitzar pantalla

5.4.5.03. Activar dispositiu



Il·lustració 22. Cas d'ús Activar dispositiu

5.5. Model conceptual de dades

Aquesta secció està dedicada a mostrar, de forma gràfica, com interaccionen entre sí cadascun dels diferents components interns que conjuntament donen funcionalitat al projecte.

Ja que es tracta d'un cas atípic (doncs es tracta de realitzar una interfície especialitzada, i per tant no disposa d'un model de dades clàssic), el que es cercarà en aquesta part del document és presentar al lector de la manera més entenedora possible com es divideix el projecte, quins components es responsabilitzen de cada divisió i com s'interrelacionen entre ells. Allà on sigui possible, el disseny en UML explicitarà també quins atributs i funcions té l'objecte representat.

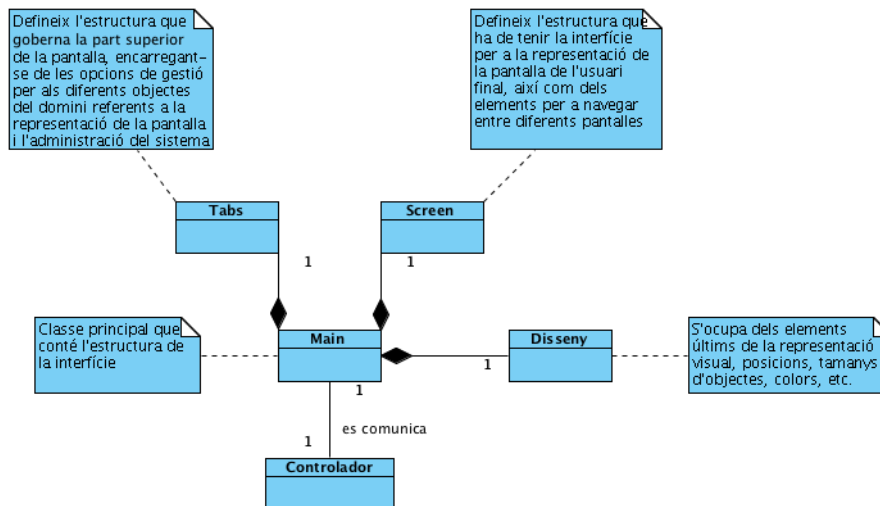
Així doncs, podem representar el model conceptual segons dos punts de vista:

- Si es consideren les tecnologies implicades, obtindríem 4 grans divisions: objectes XUL per representar l'estructura, tecnologia CSS per donar-li forma, Javascript per dotar-la de contingut i Java per comunicar-se amb els diferents elements domòtics.
- Per contra, si es té com a punt de divisió la funcionalitat que representen dins del projecte, aleshores podríem considerar 3 divisions: l'encarregada de la vista (XUL, CSS i la part del Controlador de Javascript), el Controlador (Controlador i classes auxiliar, totes en Javascript juntament amb els objectes Java de comunicació amb els dispositius domòtics) i per últim el model de dades (també totes en Javascript).

De cara a la representació dels diferents diagrames, s'ha optat per la segona forma de representació, ja que es considera la més pròxima al nivell conceptual, descartant les tecnologies implicades; tal i com correspondria a un model conceptual de dades tradicional.

5.5.1. Vista

Tal i com s'ha explicat, en aquesta subsecció es mostrarà com interactuen entre si els diferents elements que s'encarreguen de la definició de la interfície:

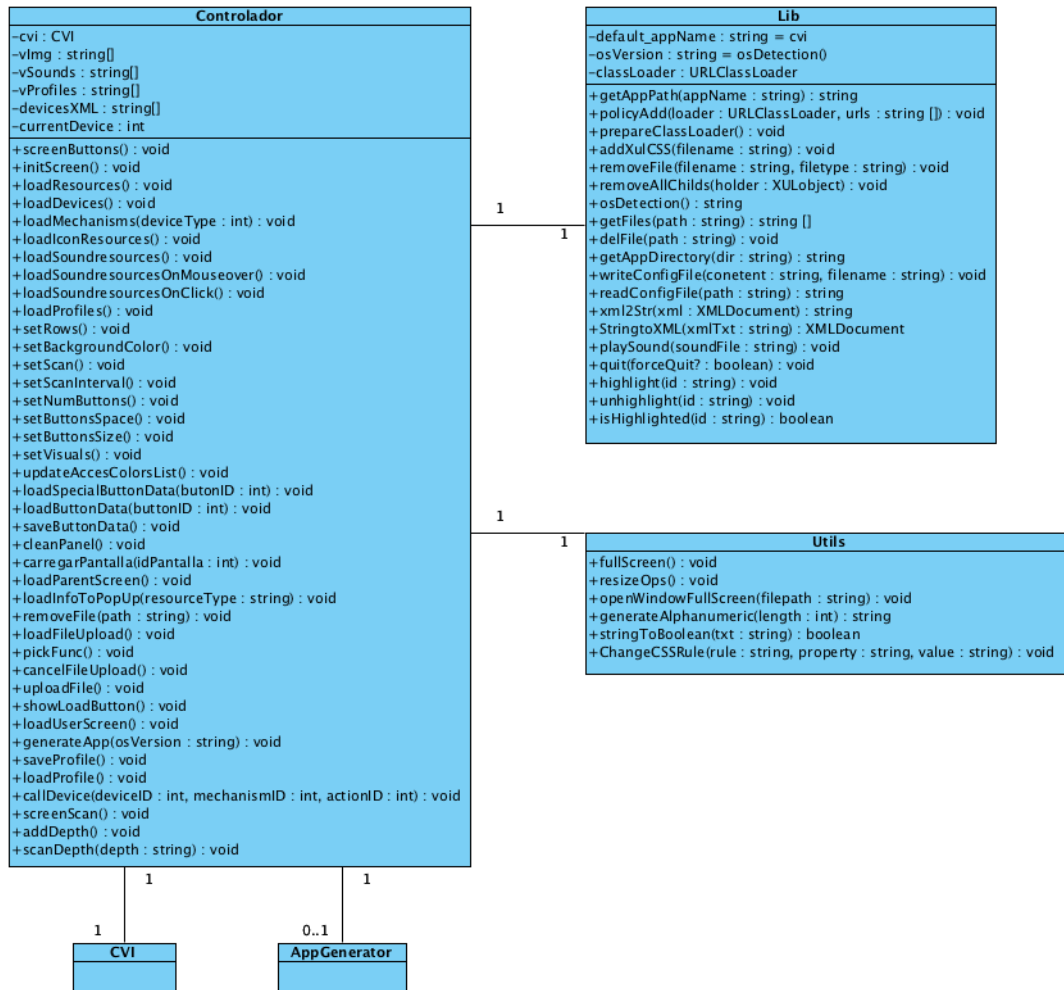


Il·lustració 23. Model conceptual referent a la vista de la interfície gràfica

Donat que aquest model no està representant objectes, no es poden classificar amb els paràmetres habituals tals com atributs i funcions. Tot i això el diagrama serveix per clarificar com s'estructuren els diferents components que defineixen la interfície gràfica.

5.5.2. Controlador

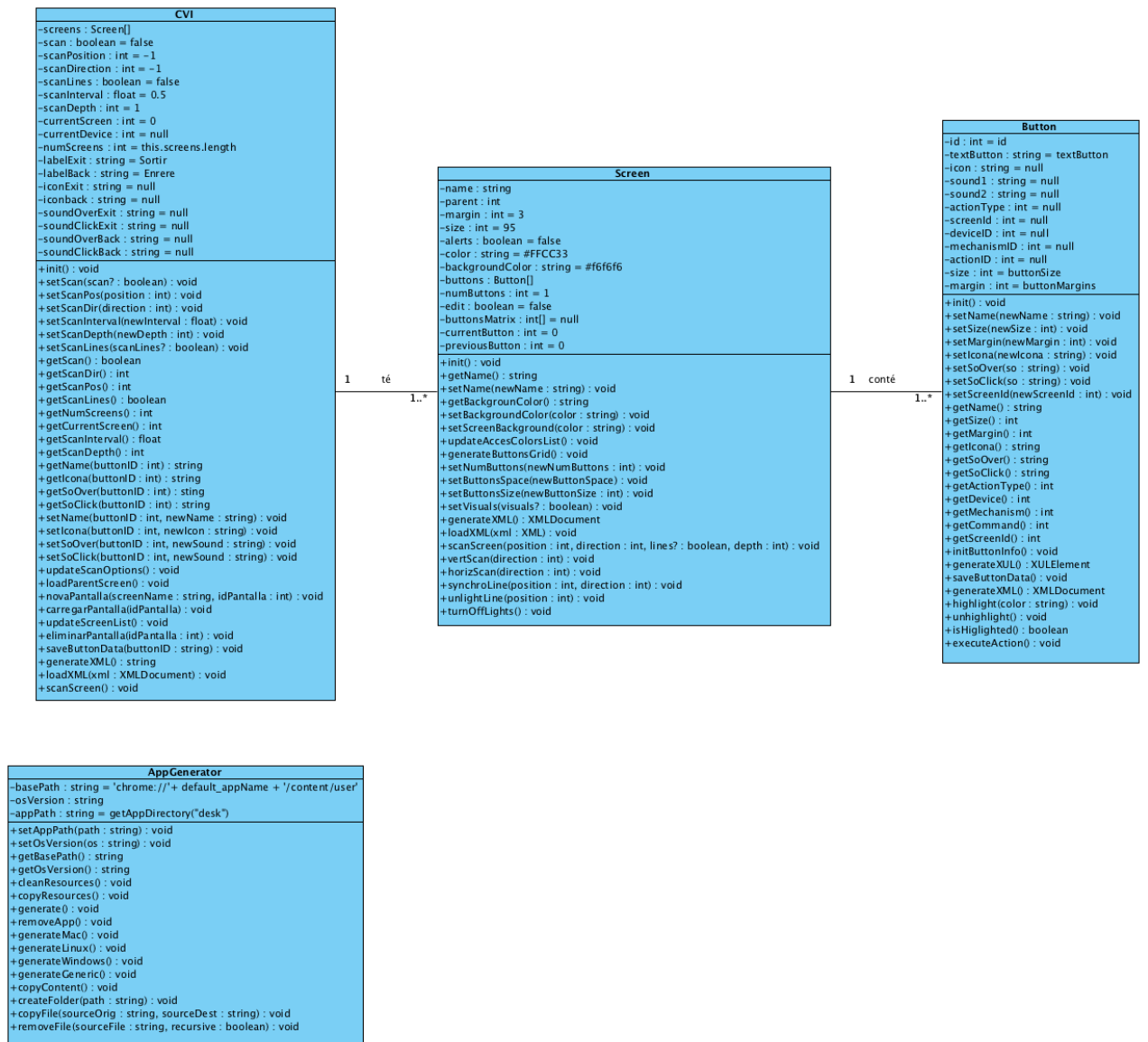
Les classes representades a continuació s'encarreguen de rebre els events capturats a la capa superior (la vista de la interfície gràfica) i tenen com a objectiu el processat d'informació i l'enllaç amb la resta d'objectes que representen l'estat del sistema.



Il·lustració 24. Model conceptual referent al controlador de la interfície gràfica

5.5.3. Domini

El Domini inclou tots aquells objectes que representen l'estat de l'aplicació.



Il·lustració 25. Model conceptual referent al domini de la interfície gràfica

6. Disseny

En la fase anterior, la fase d'Especificació, s'ha detallat què ha de fer el sistema, però falta per veure com ho fa. Per aquesta raó, en aquest capítol de Disseny es detallarà i refinarà l'especificació per a obtenir un disseny i com es duran a terme les funcionalitats ja especificades.

El disseny sempre va lligat a l'arquitectura i tecnologia que s'empraran (llenguatges de programació, plataforma, tipologia, etc.) i considera, alhora, els requisits no funcionals.

6.1. Disseny de l'arquitectura

Al tractar-se d'una aplicació que només ha d'exercir els elements d'una interfície gràfica, s'han considerat adaptacions de patrons de programació clàssics modificats per a resoldre la casuística inherent al projecte.

Així doncs, patrons com el model-vista-controlador s'han vist alterats, tot i que en essència els seus conceptes s'han respectat tant com ha sigut possible.

Si en una aplicació clàssica s'usa aquest patró per separar les dades de la interfície i de la lògica de control en tres components diferenciats, en aquest projecte s'ha adaptat cada capa a les necessitats del mateix de la forma següent:

6.1.1. Vista

Aquesta capa correspondria de forma natural a la "Vista" en el patró *mvc*. Com a tal, s'encarrega de presentar el model del sistema en un format òptim per a la seva interacció; es tracta de les classes i fitxers responsables de la interfície gràfica.

En el sistema està representada per les tecnologies XUL i CSS.

6.1.2. Controlador

Aquesta capa respon a esdeveniments, normalment peticions de l'usuari, i invoca peticions al domini; també s'encarrega del camí contrari de les dades, ja que sincronitza els canvis en el model de dades amb la seva representació en la capa Vista.

La tecnologia Javascript és l'encarregada d'establir les funcions del controlador.

6.1.3. Domini [JS + arxius XML + binaris]

Es tracta de la capa que representa amb el que el sistema opera. En general es limita a tot allò relatiu a informació provinent de la vista i del controlador, facilitant presentacions visuals complexes. El sistema també pot operar amb dades no relatives a la presentació, fent ús d'altres lògiques de negoci i dades afins amb el sistema modelat per encarregar-se de la persistència de l'estat de l'aplicació, així com de la generació d'arxius binaris per a generar aplicacions client en el format natiu del S.O. objectiu.

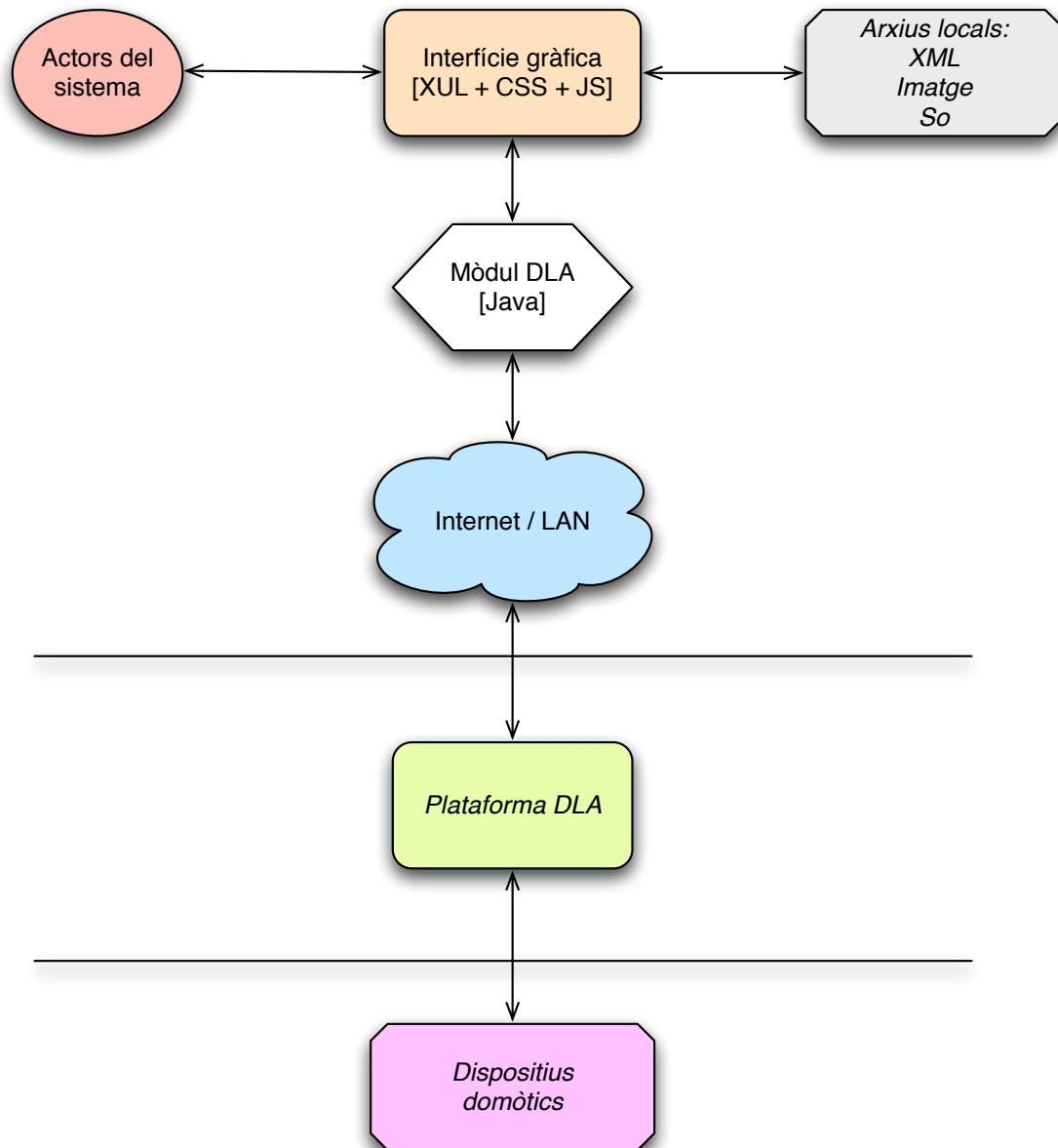
Seria l'equivalent a la capa 'Model' en el patró *mvc*, tot i que cal remarcar que en aquest cas es manté exclusivament la informació relativa a la interfície, deixant al mòdul DLA el tractament idoni per a la gestió i correcta manipulació de les dades provinents dels dispositius domòtics.

Aquestes funcionalitats es troben distribuïdes entre la tecnologia Javascript, XML i arxius binaris necessaris per a la generació d'aplicacions client.

6.2. Arquitectura física del sistema

L'arquitectura física del sistema ens permet visualitzar com es distribueixen els components lògics en el sistema en el sistema físic.

El següent diagrama ajudarà a entendre com interactuen entre sí cadascun dels elements de la interfície gràfica en el conjunt total del Sistema per a controlar una casa domòtica, i ajudarà a tenir una visió global de cadascun dels components del sistema.



Il·lustració 26. Diagrama distribució física del sistema

Com es pot observar, l'arquitectura física està dividida en tres parts:

- Part local (en l'ordinador del usuari del sistema).
Aquesta part consta dels actors responsables de l'ús del sistema i definits anteriorment en aquest mateix document.
Els actors interactuen amb la interfície gràfica, motiu del projecte que ens ocupa, i que consta de les tecnologies ja esmentades: XUL + CSS per a la definició de la Vista, Javascript per a definir el comportament de la mateixa i interactuar amb la vista i altres elements que defineixen el domini.
Recursos locals, en forma de fitxers de configuració, d'imatge i so, que el sistema necessita per definir elements complexos de la vista.
Mòdul Java per a enllaçar la interfície gràfica amb els components domòtics de la casa, mitjançant el component DLA, definit a l'inici del document.

- Part servidor (en remot o en local, segons es configuri)
Aquesta part serveix d'enllaç entre la interfície i els dispositius i s'encarrega de manipular cadascun dels diferents elements domòtics de la casa i centralitzar el control en un únic element. Per la seva complexitat, aquest element es de per sí un altre PFC independent, ja que no només requereix mapejar les direccions de dispositius domòtics, sinó també crear el hardware necessari per a que els dispositius es comuniquin amb el software que els centralitza.
- Dispositius físics.
Finalment, la última zona del sistema la engloba tots aquells dispositius domòtics que es tenen instal·lats a la casa

7. Disseny gràfic

Fins aquí s'ha explicat que ha de fer el sistema, com ho fa i de quina manera interacciona la interfície amb els altres elements del projecte global que és controlar una casa domòtica. Però s'ha de recordar que aquest projecte té un requisit anòmal si es compara a la resta de projectes informàtics, i és que ha d'arribar a un conjunt de la població amb dificultats cognitives i/o motrius.

Per aquest motiu, pren una especial importància explicar com està distribuïda visualment l'aplicació, i justificar algunes de les decisions preses alhora d'interactuar amb la interfície.

7.1. Elements clau de la interfície

En la fase de captació de requisits del projecte ja van sorgir aquells requisits no funcionals que afectaran el disseny visual de la interfície. Segurament, el més important de tots els detectats és el d'usabilitat, ja que un mateix requisit ha afectat de varies formes el desenvolupament del sistema.

Cal tenir en compte, per una banda, que els usuaris que fan ús de l'aplicació d'administració (que permet configurar el sistema per als usuaris finals), no seran gent familiaritzada amb aplicacions d'aquest estil excepte en algun cas, de manera que es molt important fer menús clars, espais lliures d'elements innecessaris i organitzar els elements de la manera més lògica possible.

D'altra banda, els usuaris finals del sistema seran persones amb dificultats cognitives o motrius. Això ha significat dues coses:

- Primerament, que s'ha de crear un model de representació visual prou adaptable com per a que serveixi per a aquests dos tipus de casuístiques, en les seves variants més complicades. El que implica això és que es necessita aportar una àmplia flexibilitat tant pel que fa a aspectes de visualització (mida, contrastos de colors..) com a aspectes d'accés a les opcions del sistema (ratolins, polsadors.. el sistema s'ha d'adaptar a diferents mètodes d'accés).
- En segon terme, també s'han considerat altres elements no tant obvis i que només un expert pot aportar. Ens referim a la capacitat que poden tenir certs tipus d'usuaris per a interaccionar amb el sistema: quantitat d'elements en pantalla i profunditat en quant a la nombre de pantalles per a navegar fins a una certa funcionalitat. Tot això també s'ha considerat al definir la interfície gràfica.

- Per últim, també cal destacar que una bona usabilitat en aquest context implica poder visualitzar prèviament i de forma ràpida el resultat final d'allò que s'està configurant, per garantir que el resultat és l'esperat.

Amb totes les consideracions pel que fa a requisits no funcionals, s'ha desenvolupat una interfície per als usuaris de tipus 'Administradors' que es divideix en tres zones principals (tal i com es comprovar en la imatge) . El motiu pel qual s'ha dividit la interfície gràfica en zones es per a facilitar l'accés a les funcionalitats, ja que agrupant totes aquelles funcionalitats que són comunes entre sí pel que fa a la seva finalitat, es busca fer un entorn de treball més còmode per a l'usuari, on s'afavoreixi la capacitat d'usar el programa per primer cop reduint tant com sigui possible l'ús de manuals o l'ajuda d'un expert.



Il·lustració 27. Distribució espai en interfície gràfica

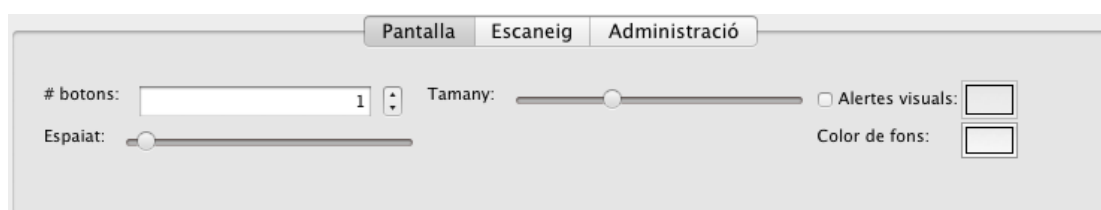
A continuació es detallen cadascuna de les zones que componen l'aplicació d'administrador:

7.2. Zona de configuració

Està situada a la part superior de l'aplicació ja que es necessita d'un espai llarg que no es solapi amb facilitat amb altres elements de la interfície en el moment que es decideixi ampliar les funcionalitats de configuració.

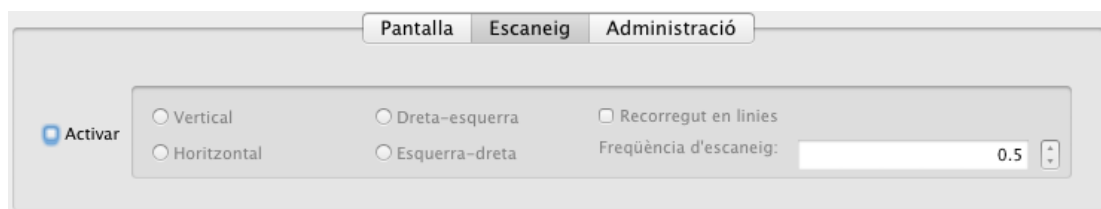
Aquesta zona presenta un format en pestanyes per reunir totes aquelles funcionalitats comunes entre sí i d'aquesta forma facilitar la cerca de funcionalitats que ofereix la interfície gràfica.

La primera pestanya s'encarrega de totes aquelles opcions referents a la configuració de la pantalla. Inclou elements tals com la definició de colors i aspectes visuals de mida, espai i nombre de botons a aplicar a la pantalla actual.



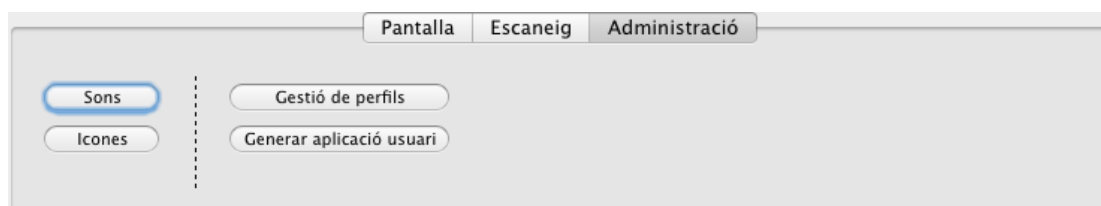
II·lustració 28. Configuració de pantalla

La pestanya d'escaneig, com el seu nom indica, permet configurar l'aplicació per a fer ús de les opcions d'escombrat automàtic dels icones de la pantalla. Aquesta funcionalitat es correspon amb la necessitat d'usuaris amb mobilitat reduïda i que només poden fer ús, per exemple, d'un polsador.



II·lustració 29. Configuració escaneig

Finalment, l'última pestanya permet configurar altres aspectes més avançat de l'aplicació: des d'aquí es poden gestionar els recursos de l'aplicació (imatges i sons), així com guardar l'estat actual per a futurs usos i fins i tot generar l'aplicació per a l'usuari final, un cop es consideri finalitzada la configuració.



II·lustració 30. Administració aplicació

7.3. Zona de navegació

Aquest espai es troba a la dreta de l'aplicació ja que la seva principal funcionalitat és la de permetre navegar entre les diferents pantalles que es configuren, així com també oferir la possibilitat de visualitzar el resultat actual sense la necessitat de generar l'aplicació per a l'usuari final.

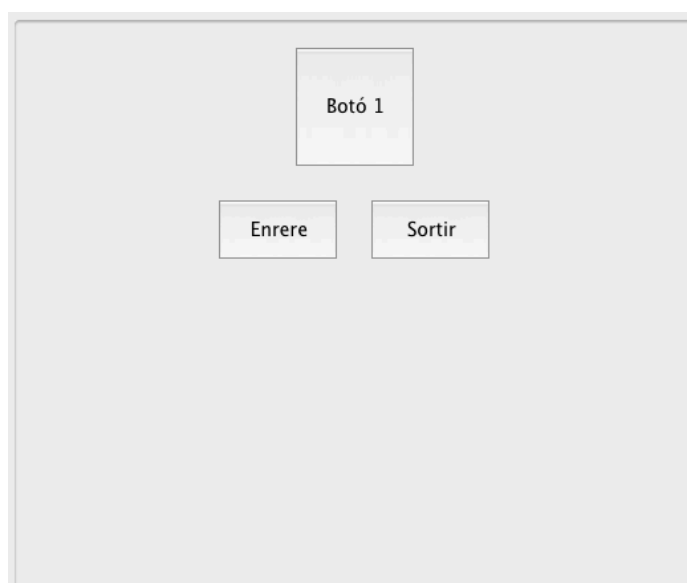


Il·lustració 31. Navegació entre pantalles

El motiu pel qual s'ha decidit usar un mètode d'organització de pantalles basat en llistats és degut a la capacitat organitzativa que aquestes suposen en contraposició a altres mètodes. En concret, en les fases d'Storyboarding es van contemplar altres mètodes (com associar una pestanya a cada pantalla existent), però es van descartar ja que no oferien un bon control de l'estat del sistema si el nombre de pantalles creixia per sobre dels límits establerts per la pantalla.

7.4. Zona de previsualització

Finalment, arribem a la última zona de l'aplicació d'administrador, que comparteix aparença amb la versió per client, ja que es tracta de l'espai on s'observa com afecten les diferents configuracions al resultat final. Canvis en la pantalla com la mida, espaiat i colors, queden reflectits instantàniament; mentre que altres canvis com les opcions d'escaneig queden inhabilitades en aquesta zona per evitar conflictes durant el període de configuració del sistema. Tot i que s'activen tant al generar l'aplicació final com al usar el botó 'Vista prèvia'.



Il·lustració 32. Previsualització pantalla. Configuració per defecte

La configuració que es carrega al iniciar l'aplicació (veure il·lustració superior) incorpora dos botons per defecte, existents a totes les pantalles, que permeten navegar i sortir del sistema.

Tanmateix, al prémer sobre un botó qualsevol, es mostren les opcions de configuració del mateix. Aquestes opcions inclouen el text associat al botó, la seva imatge, els sons al prémer-lo o seleccionar-lo i l'acció a realitzar al prémer el botó, que pot ser o bé la creació d'una nova pàgina o la realització d'una acció en un dels dispositius domòtics que consten en el DLA.

The image shows a configuration dialog box for a button. It contains the following elements:

- Text:** A text input field containing "Botó 1".
- Icona:** A dropdown menu with "Casa" selected.
- So al passar per sobre:** An empty dropdown menu.
- So al clicar:** An empty dropdown menu.
- Radio buttons:** "Nova pagina" (unselected) and "Accio" (selected).
- Accio options:** Three dropdown menus with "Llums", "rebedor", and "Encendre" selected.
- Buttons:** A "Guardar" button at the bottom.

Il·lustració 33. Configuració botó

Inicialment aquesta opció estava lligada a la Zona de configuració, però les reunions amb el CVI van desestimar la idea original, ja que la configuració canviava al prémer cada botó existent, i tenir un menú estàtic no contribuïa a associar el requadre d'opcions amb el botó que s'estava configurant. Finalment es va optar per crear un panell flotant que es situa a l'alçada del botó que s'ha premut i apareix al prémer el botó i desapareix al guardar o clicar fora del quadre (equivalent a cancel·lar).

8. Implementació

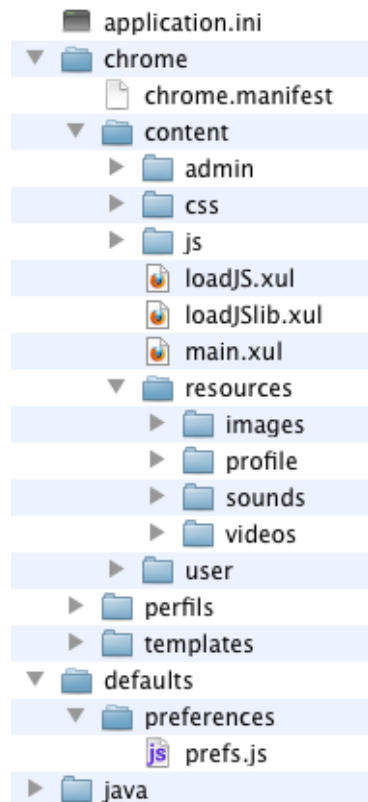
Aquesta secció del document està dedicada a descriure algunes de les parts més rellevants de l'aplicació. Per abordar aquesta tasca, el document es dividirà en quatre parts, cadascuna d'elles pertanyent a components essencialment diferents del sistema i que aporten, per tant, una funcionalitat única al context de l'aplicació. Es mostraran els fragments de codi que corresponguin a cada funcionalitat acompanyats dels comentaris pertinents en cas que sigui necessari remarcar algun aspecte. També s'explicarà, si es dóna el cas, tots aquells mètodes que sorgeixin .

8.1. Directori del Sistema

La tecnologia XUL amb la que s'ha generat la interfície presenta uns requisits alhora d'estructurar els fitxers d'execució que han propiciat que les dades a emmagatzemar s'adaptessin a aquest sistema.

Per a realitzar la persistència de dades del sistema, es fa ús dels mètodes de conversió a bytes d'objectes que aporta el llenguatge de programació per així escriure'ls i/o llegir-los de disc. Per a la gestió d'aquests tipus d'arxius s'ha creat un conjunt de directoris a fi d'emmagatzemar i organitzar també altres recursos com les imatges i els sons dels que ha de fer ús l'aplicació.

A continuació, es mostra de forma esquemàtica com es distribueixen els fitxers:



Il·lustració 34. Directori de fitxers

Tal i com es pot apreciar en la imatge, una aplicació feta en XUL presenta una jerarquia de fitxers.

A l'arrel de l'estructura es troba els fitxers *Application.ini*, que descriu l'aplicació i el seu entorn; i les carpetes *chrome* (on s'emmagatzema els fitxers de l'aplicació), *defaults* (que defineix les variables d'execució de l'aplicació) i *Java*, carpeta aliena a l'estructura base d'una aplicació XUL que gestiona totes aquelles funcionalitats del sistema que son dependents de la tecnologia Java (principalment la connexió amb el DLA).

Dins de la carpeta *chrome*, es defineixen:

- *chrome.manifest*: Aquest fitxer defineix el camí al fitxer principal que executa l'aplicació; en altres llenguatges, es definir el camí fins al document que conté la funció Main.
- *Content*: Carpeta que emmagatzema els arxius en format XUL, Javascript i CSS de l'aplicació. També s'usa per guardar aquells continguts extra, com els recursos de l'aplicació (sons, imatges i vídeos).

- *Perfils*: En aquesta carpeta es guarda de forma persistent l'estat de l'aplicació per a una posterior càrrega. També serveix com a carpeta temporal per a guardar l'estat actual de l'aplicació en el moment de generar una aplicació d'usuari.
- *Templates*: Es l'última carpeta que guarda elements referits al contingut de l'aplicació. En concret, emmagatzema les peculiaritats de cada plataforma: motor d'execució XULRunner, estructura de fitxers pròpia a cada Sistema Operatiu i altres configuracions adaptades.

8.2. Inicialització aplicació

Cada cop que s'inicia l'aplicació, ja sigui en la versió 'administració' com en la versió 'usuari', s'han d'inicialitzar els elements que hi ha en pantalla, així com també es necessari obtenir els recursos existents en el sistema per a futurs usos. Aquesta càrrega inicial de recursos pot arribar a augmentar el temps de càrrega de l'aplicació si es disposa d'una àmplia quantitat de recursos, tot i que disminueix el temps d'accés posterior quan es vulguin usar els recursos ja que només es realitza la càrrega una única vegada.

El procés que segueix l'aplicació és el següent:

- Inicialment es llegeix l'arxiu '*prefs.js*', que presenta el següent contingut:

```
pref("toolkit.defaultChromeURI", "chrome://cvi/content/main.xul");
/* debugging prefs */
/*
pref("browser.dom.window.dump.enabled", true);
pref("javascript.options.showInConsole", true);
pref("javascript.options.strict", true);
pref("nglayout.debug.disable_xul_cache", true);
pref("nglayout.debug.disable_xul_fastload", true);
pref("dom.allow_scripts_to_close_windows", true);
*/
```

Il·lustració 35. Fitxer de preferències

Aquest fitxer indica quin és el fitxer per defecte a carregar. També es pot usar per definir altres variables d'execució, com opcions per facilitar el 'debugging' durant l'etapa de desenvolupament de la interfície.

- En el fitxer en XUL '*main.xul*', s'importen tots aquells recursos que seran necessaris pel sistema: arxius Javascript amb la lògica de l'aplicació, arxius XUL amb l'esquelet del sistema gràfic i arxius CSS també, que expliciten l'aparença d'alguns elements gràfics de forma estàtica.

```

<?xml version="1.0" encoding="utf-8"?>
<?xml-stylesheet href="chrome://global/skin/" type="text/css"?>
<?xml-stylesheet href="css/default.css" type="text/css"?>

<?xul-overlay href="admin/tabs.xul"?>
<?xul-overlay href="admin/screen.xul"?>

<?xul-overlay href="loadJS.xul"?>
<?xul-overlay href="loadJSlib.xul"?>

<window
  id          = "mainWindow"
  xmlns:html = "http://www.w3.org/1999/xhtml"
  xmlns      = "http://www.mozilla.org/keymaster/gatekeeper/
there.is.only.xul"
  title      = "UPC - FiB :: generador"
  orient     = "vertical"
  onload     = "initScreen();"
  onresize   = "setRows()"
  screenX    = "0"
  screenY    = "0"
  sizemode   = "maximized"
>
<vbox>
  <hbox id="tabs" />

  <splitter width="0" collapse="before" resizeafter="farthest"
oncommand="setRows()" >
    <grippy style="width=100%;"/>
  </splitter>

  <hbox id="content">
    <panel id="button-config" />
  </hbox>

</vbox>
</window>

```

Il·lustració 36. Definició de l'esquelet de l'aplicació

Tal i com s'observa en la imatge, l'aparença d'un arxiu XUL es molt similar a la d'altres llenguatges derivats del XML, com l'XHTML per exemple, tot i que els tags són d'ús propi.

- Amb l'estructura definida, es necessita dotar de dades la mateixa, i per aquesta raó el main crida la funció `initScreen`:

```

function initScreen() {
  //assignar permisos que permetin llegir cadascun dels fitxers emmagatzemats en el sistema
  netscape.security.PrivilegeManager.enablePrivilege("UniversalXPConnect");

  //1. carregar recursos de disc (imatges, icones..)
  loadResources();
  //2. carregar objectes (pantalles, botons...)
  var pos = cvi.currentScreen;
  cvi.init();
}

```

Il·lustració 37. Càrrega inicial del sistema

Per a que aquesta càrrega sigui satisfactòria, es necessiten declarar prèviament un conjunt de variables d'entorn globals que permetin al Controlador interactuar amb la resta de classes que conformen la lògica del sistema.

8.3. Persistència de fitxers

Pel que fa al tractament de les dades, ja s'ha comentat que aquestes s'emmagatzemen a disc tant pel que fa a l'estat del sistema com dels recursos emmagatzemats.

Durant la càrrega inicial es fan lectures de disc de les dades del sistema. Una lectura qualsevol es realitza de la següent forma:

```
function getFiles(filePath)
{
    try {
        var file = Components.classes["@mozilla.org/file/local;1"]
            .createInstance(Components.interfaces.nsILocalFile);
        file.initWithPath(filePath);

        // file is the given directory (nsIFile)
        var entries = file.directoryEntries;
        var array = [];

        while (entries.hasMoreElements()) {
            var entry = entries.getNext();
            entry.QueryInterface(Components.interfaces.nsIFile);

            if (entry.isFile()) array.push(entry);
        }

        return array;
    } catch(e) { alert(e + ", on line: " + e.lineNumber); }
}
```

Il·lustració 38. Lectura de dades de disc

Amb aquest mètode es compleix l'objectiu d'obtenir els fitxers de disc donat un camí absolut, que prèviament s'ha creat de forma dinàmica.

Pel que fa a la creació i eliminació de documents físics, no presenten grans canvis i per tant no mereixen una menció especial en el document, tot consisteix en crear un element del tipus nsIFile (tal i com es reflecteix a la imatge) i invocar mètodes com *'remove'* (per eliminar), *'create'* per crear-lo (en aquest cas seria un fitxer buit que s'hauria d'omplir de contingut amb altres mètodes) o *'copyTo'* per copiar un fitxer.

Des del punt de vista de la implementació, el concepte més interessant rau en l'ús d'elements XPCOM (*Components.classes["@mozilla/file/local;1"]* en la il·lustració anterior) per a accedir als fitxers. Es tracta de components multiplataforma orientats a objectes; aquest components es poden escriure en llenguatges com C++ i es poden integrar al motor d'execució XULRunner, tot i que la configuració estàndard del motor ja porta prou elements XPCOM per a tractar amb la majoria d'elements bàsics.

8.4. Connexió DLA

Durant la inicialització de l'aplicació no només s'han d'inicialitzar aquells elements que fan ús dels recursos del sistema, doncs una part molt important tracta l'enllaç amb el sistema DLA que controla els dispositius domòtics de la casa.

```
var basePath = getAppPath(default_appName);
var urlArray = new java.net.URL(basePath + 'java/');
classLoader = new java.net.URLClassLoader( [ urlArray ] );
policyAdd(classLoader, urlArray);

//Cridem a la classe Java corresponent
var aClass = java.lang.Class.forName("domoticDevices", true, classLoader);
var inst = aClass.newInstance();

var fileName = getAppDirectory('defaults') + '/preferences/dla.ini';
fileName = adaptPathToOS(fileName);

inst.setFileName(fileName);
inst.init();
```

Il·lustració 39. Crida a codi Java (DLA) des de Javascript

En la il·lustració es pot observar com s'inicialitzen components Java i es criden a les seves classes des de Javascript. Concretament es pot observar la inicialització del sistema que permetrà comunicar-se amb el DLA.

Per a realitzar la connexió, es necessita primer de tot el servidor funcionant i visible a la xarxa, amb una direcció coneguda que permeti establir un enllaç entre la interfície i el servidor. Amb els sistemes funcionant, només cal establir l'enllaç tal i com es mostra a continuació:

```
//creacio objecte de lectura
ReadDevicesXML xmlDLA = new ReadDevicesXML();
//preparacio dades a enviar
byte[] data = xmlDLA.toByteArray();
//posicio inicial de lectura de dades
int pos = 0;
//connexio amb el DLA
this.NewConnection = DLA.NewConnection(host, port, "ReadDevicesXML", xmlDLA.getSize());
```

Il·lustració 40. Connexió amb el DLA

Amb la connexió establerta, ja es pot enviar i rebre informació entre el servidor i la interfície gràfica. En l'exemple actual, cal obtenir els dispositius disponibles. Per a aquesta tasca es necessita un bucle, doncs s'ha de tenir en consideració que el DLA només accepta treballar amb conjunts de dades fixes, tot i que es permet crear estructures de dades pròpies que es converteixen a bits de cara al enviament d'informació entre servidor i client. Donat que el nombre de dispositius és variable i la mida de bytes per realitzar la connexió ha d'estar fixada, s'ha decidit optar per un mètode d'enquesta que permet obtenir les dades a trossos; després la lògica de la interfície s'encarregarà d'ajuntar les dades de nou per obtenir el resultat demanat.

```

//comprobacio final de la transmissio
while((int)xmlDLA.getEnd() != 1) {
    xmlDLA.setBegin((byte) 0); //indicar que no es vol iniciar el programa
    xmlDLA.setNext((byte) 1); //indicar que es vol la següent part
    data = xmlDLA.toBytes();
    //indiquem al servidor que volem llegir la següent part del fitxer
    DLA.OutputConnection(NewConnection, data, pos, xmlDLA.getSize());
    //mentres el servidor no hagi escrit, fer enquesta
    while(xmlDLA.getNext() == 1){
        //Lectura estat del servidor
        DLA.InputConnection(NewConnection, data, pos, xmlDLA.getSize());
        //conversio a objecte
        xmlDLA.toObject(data);
    }
    //concatenacio xml parcial al xml general
    xml += new String(xmlDLA.getXml());
}

```

Il·lustració 41. Codi enquesta fitxer XML al DLA

A nivell general, els mètodes necessaris per a realitzar peticions des de la interfície cap al DLA es realitzen amb la mateixa metodologia que l'exemple mostrat, adaptant les estructures que cal usar al tipus de dades que es transmeten.

Un cop es finalitza l'execució del codi Java, només queda processar el resultat de nou en la interfície gràfica, com es pot observar a continuació:

```

var devicesText = inst.getDevices();

//clean xml from unknown sources
var end = devicesText.lastIndexOf(">");
devicesText = devicesText.substring(0, end+1);

devicesXML = StringtoXML(devicesText);

```

Il·lustració 42. Javascript processant dades retornades via Java

8.5. Generació d'aplicacions

Fins ara s'han observat parts de codi que tracten des de la inicialització de l'aplicació fins a la connexió amb el servidor gestor dels dispositius domòtics, passant pels mètodes d'accés a dades persistents.

Encara falta per comentar, però, el sistema usat per a accedir a l'aplicació de l'usuari final.

En aquest aspecte cal remuntar-se a la necessitat inicial: aconseguir que gent amb algun tipus de problema motriu o cognitiu (a qui també s'anomena en el seu conjunt com 'usuaris'), puguin connectar-se a diferents elements domòtics de la casa de forma accessible amb la finalitat d'intentar contribuir a la millora de la qualitat de vida d'aquest col·lectiu d'usuaris.

Per a assolir l'objectiu, s'ha creat una interfície que es comunica amb els dispositius domòtics i permet establir les línies generals de com ha de ser la interfície que l'usuari utilitzarà, així com el comportament esperat de cadascun dels elements d'aquesta.

Però per a que l'usuari faci ús d'aquesta configuració, ha de tenir una aplicació pròpia que es centri de forma exclusiva en la interfície generada, eliminant tot rastre de les opcions d'administració.

Així doncs, es van proposar diferents alternatives que ajudessin a l'objectiu:

- Inicialment, la opció clàssica és la d'establir '*logins*', de manera que una mateixa aplicació pot comportar-se de maneres diferents en funció de l'usuari que hi accedeix. Aquesta opció, però, es totalment ineficient quan es tracta amb el col·lectiu d'usuaris d'aquest projecte, doncs obligar a fer '*login*' dificulta en gran mesura l'accessibilitat al sistema.
- La segona proposta pretenia solucionar el problema anterior: disposar de dues aplicacions, una dedicada a l'administració i una altra que satisfà les necessitats dels usuaris finals. La idea de tenir dues aplicacions, però, suposa novament un problema, ja que s'ha de tenir un espai compartit per passar les configuracions d'una aplicació a l'altre, definir quina configuració es vol carregar per defecte i assegurar-se que no es modifiquen les rutes d'accés a aquest espai compartit, sinó l'aplicació client no podria llegir la configuració.
- Amb tots aquests problemes nous, es va decidir descartar aquesta idea i finalment es va apostar per una nova variant, que encara que és més costosa a nivell d'implementació, incorpora els beneficis de les dues propostes anteriors, tot minimitzant els compromisos a prendre. Es tracta, doncs, de fer que l'aplicació d'administració generi de forma dinàmica l'aplicació de l'usuari final en base a l'estat actual del sistema.

Per a fer-ho, s'ha vist a l'inici de l'apartat que es disposa d'una carpeta amb les dades pròpies de cada plataforma, diferents a l'aplicació d'administració i que per tant no es poden reciclar.

Tanmateix, la copia de recursos propicia generar una aplicació d'usuari que ocupa la mida mínima exigida per al funcionament òptim: no es necessari compartir espai de fitxers i tots aquells recursos que no usa la configuració final no s'inclouen en la generació de l'aplicació de l'usuari.

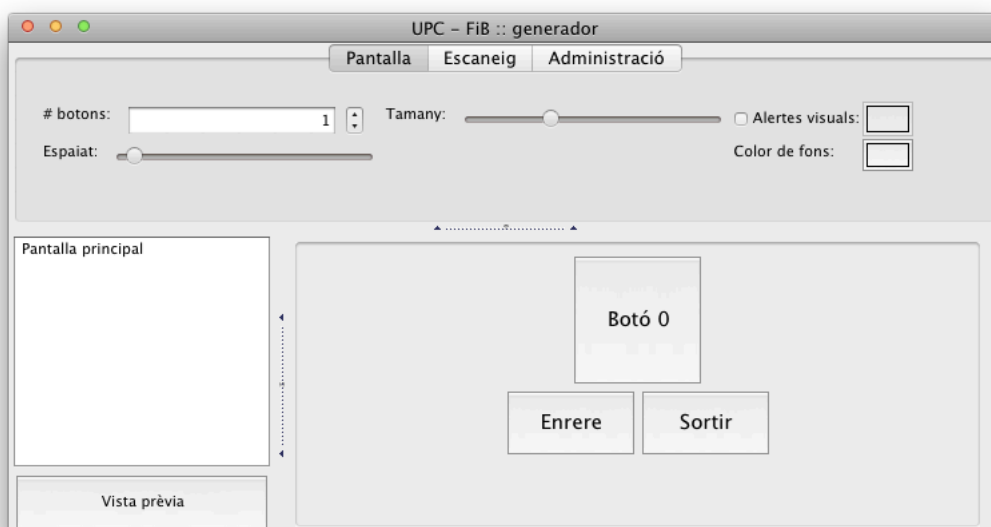
Per acabar la generació, s'ha determinat que la senzillesa en la seva creació premiava per damunt d'altres funcionalitats, de manera que s'ha optat per generar les aplicacions en l'escriptori que executa l'aplicació, d'aquesta manera sempre es podrà trobar fàcilment s'usi el Sistema Operatiu que s'usi.

8.6. Exemple d'implementació

Per acabar aquest apartat, es mostra a forma d'exemple una interacció completa entre la interfície i el sistema, amb la finalitat d'aportar al lector una comprensió global de cadascun dels elements implementats i com es transmeten informació entre ells.

Per a aquesta finalitat, s'usarà el mètode per a configurar botons, que permet observar la interacció entre XUL i Javascript.

Partint d'una interfície d'administrador ja carregada, es disposa de la vista original que ofereix el sistema. Tot i que el sistema treballa a pantalla completa, les imatges es veuran reduïdes per adaptar-les a un millor visionat en paper:



Il·lustració 43. Pantalla inicial d'interfície d'administrador

Sobre aquesta pantalla inicial, es clica sobre el botó 'Botó 0'. Aquesta acció fa que el botó activi la propietat definida durant la generació dinàmica del seu codi:

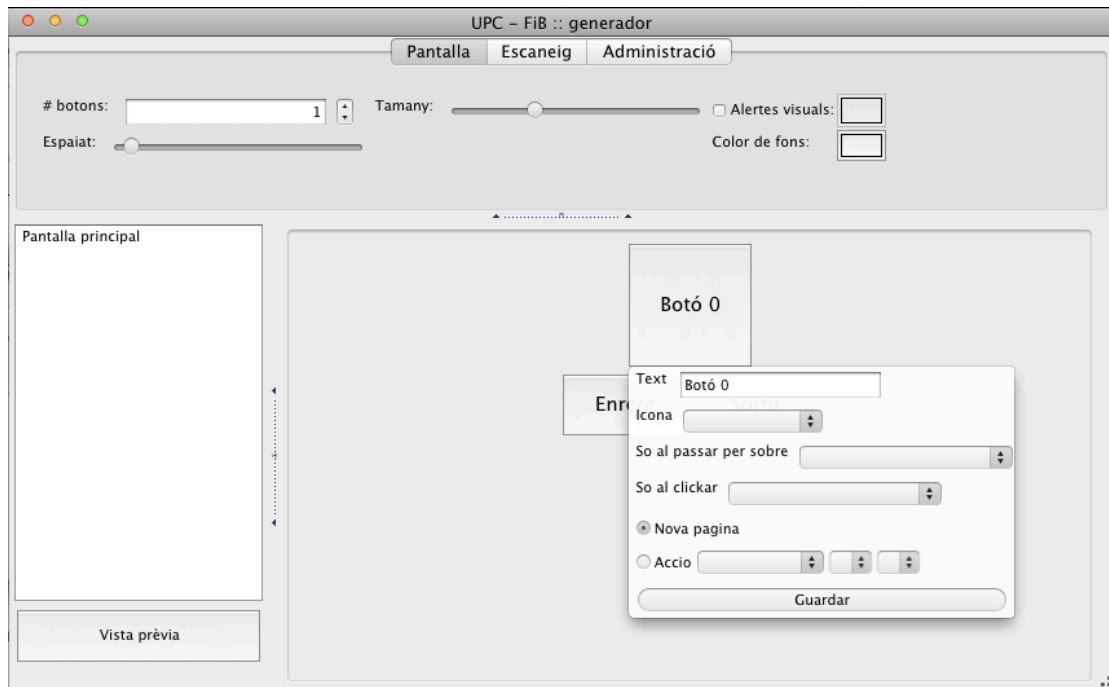
```
var onclickString = "document.getElementById('button-config').openPopup(this, 'before_start');  
loadButtonData(this.id);";
```

Il·lustració 44. Codi d'invocació Javascript a un panell XUL

Amb aquesta acció, sobre el panell definit amb el nom 'button-config'. Aquest panell s'ha definit en XUL, ja que es tracta d'una interfície, i mitjançant la funció 'loadButtonData', la lògica del sistema s'encarregarà d'afegir tota la informació relativa al boto al panell de configuració del botó.

```
<panel id="button-config" >  
  <vbox style="width:200px;">  
    <hbox>  
      <label value="Text" />  
      <textbox id="p_buttonText" />  
    </hbox>  
    <hbox>  
      <label value="Icona" />  
      <menulist id="p_icon" editable="false" />  
    </hbox>  
  </vbox>  
  
  <vbox style="width:200px;">  
    <hbox>  
      <label value="So al passar per sobre" />  
      <menulist id="p_mouseover_sound" editable="false" />  
    </hbox>  
  </vbox>  
  
  <vbox style="width:200px;">  
    <hbox>  
      <label value="So al clicar" />  
      <menulist id="p_onclick_sound" editable="false" />  
    </hbox>  
  </vbox>  
  
  <vbox>  
    <radiogroup id="p_action" >  
      <radio label="Nova pagina" value="new_page"/>  
      <hbox>  
        <radio label="Accio" value="action"/>  
        <menulist id="p_device" editable="false" oncommand="loadMechanisms(this.value)"/>  
        <menulist id="p_mechanism" editable="false" />  
        <menulist id="p_command" editable="false" />  
      </hbox>  
    </radiogroup>  
  </vbox>  
  
  <vbox>  
    <button id="guardarButtonInfo" label="Guardar" oncommand="saveButtonData();" />  
  </vbox>  
  
  <box hidden="true" id="buttonID">  
    <label id="buttonID_label" value="" />  
  </box>
```

Il·lustració 45. Definició d'un panell en XUL



Il·lustració 46. Desplegament en interfície del panell invocat per Javascript.

Es pot observar que el panell integra les opcions tant d'edició de l'aspecte del botó, com de les funcionalitats que se li assignen.

Si es deixen les configuracions per defecte i es clica guardar, s'invoca la funció *'saveButtonData'*, que enllaça l'XUL amb la lògica de l'aplicació.

Aquest mètode, que està definit en el controlador, s'encarrega de tractar la informació del panell i decidir quina classe ha de gestionar la informació adscrita, ja que cada pantalla té assignats N botons, cadascun amb la seva configuració, excepte els botons 'Enrere' i 'Sortir', genèrics a totes les pantalles i que es troben sota la supervisió de la classe CVI, encarregada de gestionar les pantalles de l'aplicació i les dades comuns a tot el sistema.

```

function saveButtonData() {
  try {
    var floatingPanel = 'button-config';
    var buttonID      = document.getElementById("buttonID_label").value;
    var pos           = cvi.currentScreen;

    //obtenir info del panel
    var label        = document.getElementById("p_buttonText").value;
    var icona        = document.getElementById("p_icon").value;
    var so1          = document.getElementById("p_mouseover_sound").value;
    var so2          = document.getElementById("p_onclick_sound").value;
    var bAction      = document.getElementById("p_action").value; // 'new page' or 'action'

    var p_device     = document.getElementById("p_device").value;
    var p_mechanism  = document.getElementById("p_mechanism").value;
    var p_command    = document.getElementById("p_command").value;

    switch(buttonID) {
      case "exitButton": case "back":
        //actualitzar info del boto
        if(label != '') {
          document.getElementById(buttonID).label = label;
          cvi.setName(buttonID, label);
        }
        cvi.saveButtonData(buttonID, label, icona, so1, so2); break;
      default:
        //actualitzar info del boto
        if(label != '') {
          document.getElementById(buttonID).label = label;
          cvi.screens[pos].buttons[buttonID].setName(label);
        }
        cvi.screens[pos].buttons[buttonID].saveButtonData(label, icona, so1, so2,
bAction, p_device, p_mechanism, p_command); break;
    }

    //tancar el panel
    document.getElementById(floatingPanel).hidePopup();

    cvi.screens[pos].generateButtonsGrid();
  } catch(e) { alert(e + ", on line: " + e.lineNumber); }
}

```

Il·lustració 47. Captura de petició per guardar les dades realitzat per la interfície al controlador

```

this.saveButtonData = function (label, icona, so1, so2, bAction, p_device, p_mechanism, p_command) {
  try {
    //funcio que es crida quan es prem al boto "guardar"
    //actualitza l'estat de les variables, aixi com també la part visual del mateix
    if(icona == -1 || icona == '' ) { this.setIcona(null); } else { this.setIcona(icona); }
    if(so1 == -1 || so1 == '' ) { this.setSoOver(null); } else { this.setSoOver(so1); }
    if(so2 == -1 || so2 == '' ) { this.setSoClick(null); } else { this.setSoClick(so2); }

    //si es nova pagina, crear la pagina nova
    switch(bAction) {
      case "new_page":
        this.actionType = 0;
        if(this.screenId == null) {
          //creem la nova pantalla i associem el Id corresponent
          cvi.novaPantalla(this.getName());
          //s'usa el lenght ja que marca la ID del nou element
          //(vector de 0 a n-1, amb n posicions, n es la nova posició a ocupar)
          this.setScreenId(cvi.getNumScreens() -1);
        }
        else {
          var screenID = this.getScreenId();
          var screenName = this.getName();

          cvi.screens[screenID].setName(screenName); //actualitzem el nom de la pantalla.

          //actualitzem la GUI
          cvi.updateScreenList();
        }
        break;
      case "action":
        this.actionType = 1;
        this.deviceID = p_device;
        this.mechanismID = p_mechanism;
        this.actionID = p_command;

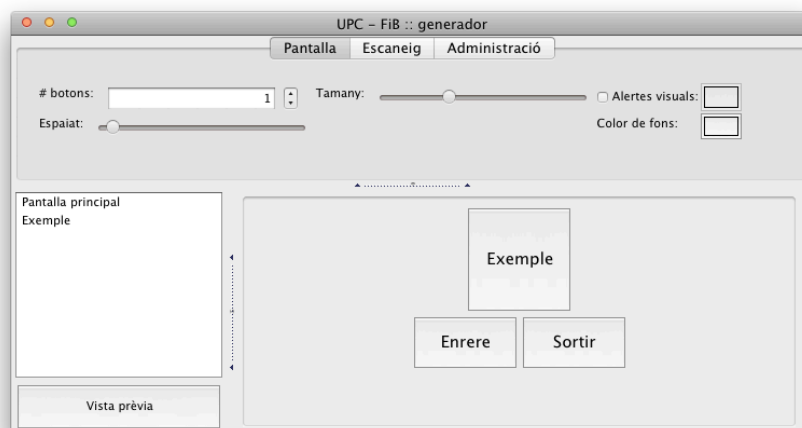
        if(this.screenId != null) {
          cvi.eliminarPantalla(this.screenId);
          this.screenId = null;
        }

        break;
    }
  } catch(e) { alert(e + ", on line: " + e.lineNumber); }
}

```

II·lustració 48. Modificació de l'estat d'un botó

El sistema actualitza l'estat del botó i, mitjançant la crida *'generateButtonsGrid'* del controlador, s'actualitza la interfície amb el nou estat del botó:



II·lustració 49. Resultat modificació botó interfície

En el cas de l'exemple, s'ha modificat el nom del botó i, al deixar la resta de les opcions per defecte, s'ha assignat l'acció del botó a obrir una nova pàgina, tal i com queda reflectit en el quadre lateral esquerre.

9. Proves

Les proves del sistema és un procés que permet comprovar, de forma manual o automàtica, la qualitat del producte, permeten identificar no només errors en la implementació (o en qualsevol etapa anterior del projecte), sinó també errors en la usabilitat i qualitat del producte.

La fase de proves d'un sistema és molt important en qualsevol projecte, ja que permet detectar qualsevol problema o defecte de funcionament, facilitant així la resolució de problemes i minimitzant els errors abans no es comenci a usar en entorns de producció. Doncs tot i que provar el sistema i solucionar errors es molt important, no es pot ignorar que l'objectiu es reduir el nombre d'errors, i no crear un sistema perfecte; partint de la base que un sistema perfecte, per definició, no pot existir.

Aquest procés de proves (o *'testing'*) ha de ser realitzat, idealment, per persones externes al projecte, doncs si un desenvolupador avalua les funcionalitats que ell mateix a creat, introdueix biaixos involuntaris; a més a més, fer que una persona aliena provi el sistema permet detectar errors o mancances que el desenvolupador original no havia pensat.

Cal comentar que les proves del sistema no les ha realitzat només el desenvolupador, sinó que també hi han contribuït gent de l'entorn del CVI, així com altres interessats, que han contribuït a trobar errors a temps i a millorar el programa en la seva globalitat.

El procés de provatura del sistema s'ha anat desenvolupant a mesura que s'implementaven cadascuna de les funcionalitats del sistema. Tot i això, no ha sigut fins a la part final del mateix que no s'han dut a terme el gran gruix de proves, ja que es requeria un sistema funcional on es poguessin provar el major nombre de variacions possibles.

Pel que fa al mètode de resolució, quan s'ha trobat una falla en el sistema, aquesta s'ha solucionat per, a continuació, tornar a executar el conjunt de proves que han fet reproduir la falla del sistema i així poder assegurar la seva correcta correcció.

Existeixen moltes metodologies pel que fa a les proves funcionals d'un sistema. A continuació es descriuran les proves que s'han usat durant el projecte:

9.1. Proves unitàries

Una prova unitària es aquella que comprova el correcte funcionament d'un bloc de codi (es a dir, d'una funcionalitat). Es un mètode que serveix per garantir que cadascuna de les funcionalitats, de forma individual, es comporta de la forma esperada.

A aquest tipus de proves les segueix gairebé sempre el conjunt de proves que integren varies funcionalitats del sistema, ja que provar només les funcionalitats per separat indueix a error: es molt comú provar les funcionalitats partint de situacions ideals (i més si es tracta del mateix desenvolupador de la funcionalitat qui fa les proves).

A més, proves d'aquest tipus no serveixen per descobrir la totalitat d'errors en el codi del sistema, de manera que un correcte ús d'aquest sistema de testeig ha d'anar seguit d'altres tipus de proves.

9.2. Proves integrals

Les proves integrals son aquelles que es realitzen en l'àmbit del desenvolupament del software un cop s'han passat totes les proves unitàries del sistema. Aquestes proves serveixen per verificar que el conjunt del sistema funciona correctament, comprovant que el comportament es l'esperat al barrejar situacions ordinàries i també casos considerats extrems.

A efectes pràctics, aquestes proves s'han realitzat quan s'ha disposat d'un sistema en un estat molt avançat, pròxim a la seva finalització.

9.3. Proves de regressió

Aquestes proves es realitzen per verificar que no s'han introduït nous errors en funcionalitats ja provades quan es realitzen canvis en el sistema.

Els mòduls que s'han vist més afectats han estat el mòdul de generació de perfils, usat tant per la interfície com pel generador d'aplicacions client, ja que es veu afectat per cadascuna de les àrees que mantenen l'estat del sistema i es corria el risc que al modificar algun comportament de l'estat del sistema, això afectes la capacitat de fer persistent l'estat del mateix, i de retruc afectes funcionalitats dependents. Això no significa, però, que no es renunciés a provar el sistema en la seva totalitat cada cop que s'han realitzat canvis en el part del sistema.

9.4. Proves d'acceptació

En essència es tracta de proves funcionals que realitza el client sobre un sistema complet i cerquen l'aprovació i completesa dels requisits detectats a l'inici del projecte.

Aquestes proves no es realitzen fins a tenir superades totes les fases de proves anteriors. No tindria sentit que el client provés tecnologia que encara està en fase de desenvolupament, o que estigues subjecte a canvis detectables per part de l'equip responsable del desenvolupament del sistema. Avançar de forma prematura aquesta fase comportaria molt probablement una pèrdua de la confiança per part del client, que es trobaria amb un sistema incomplet i possiblement amb una qualitat massa baixa.

10. Planificació del projecte

10.1. Planificació temporal

La planificació temporal implica l'organització de les diferents etapes i tasques que componen el projecte en el temps, partint de la data d'inici fins al tancament del mateix.

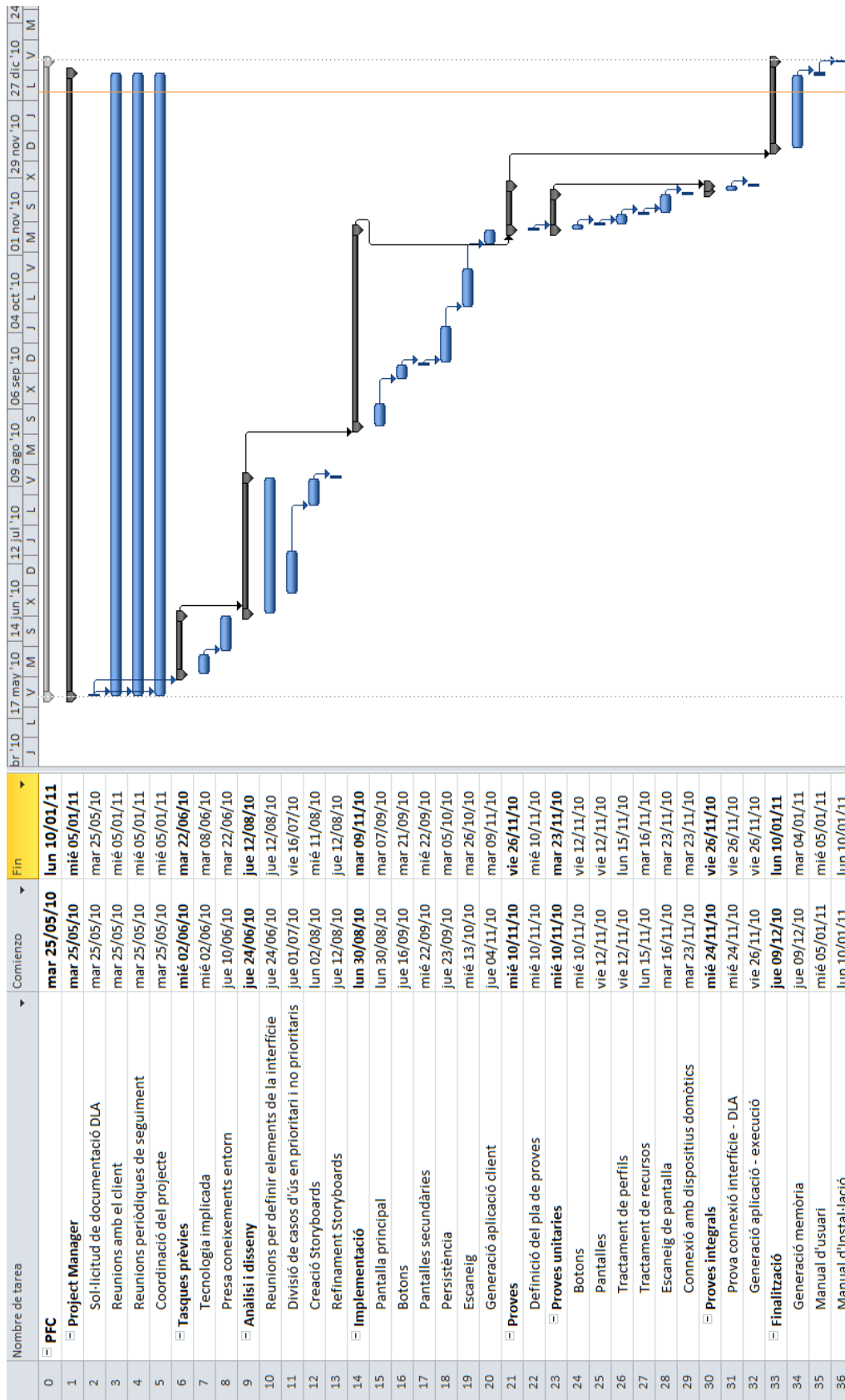
Aquest projecte es va iniciar a finals de Maig del 2010, i com qualsevol altre projecte en el món empresarial, ha patit alguns contratemps. La major part dels problemes han sorgit al compaginar el projecte amb altres assignatures de la FIB (només a l'inici del projecte) i la feina (durant tot el projecte).

Inicialment es pretenia tenir acabat el projecte per finals d'any, al considerar-se un projecte de dificultat mitjana, però un contratemps en la part final de la implementació el va demorar unes setmanes. El projecte havia de funcionar conjuntament amb el PFC d'un altre alumne que ja estava treballant en el mateix CVI amb dispositius domòtics i havia desenvolupat una software que, a baix nivell, integrava tots els elements dispers de la casa en un únic punt de control. Aquest software, però, requeria inicialment que el sistema tingués coneixement de les adreces físiques dels dispositius per a cridar-los, de manera que ha calgut adaptar el sistema, conjuntament amb l'estudiant creador del software, per a que sigui capaç de comunicar de forma automàtica els dispositius, així com també rebre ordres a més alt nivell.

Aquesta contratemps es podria haver solucionat si s'hagués experimentat amb el DLA a principi del projecte, amb el temps suficient per integrar els canvis al calendari, sense que això afectés la resta del projecte; per tant, es tracta d'una planificació incorrecta al no tenir en consideració totes les dificultats possibles.

Tot seguit es mostra un Diagrama Gantt amb el pla de treball del projecte:

10.1.1. Diagrama de Gantt



Il·lustració 50. Diagrama Gantt

10.2. Anàlisi econòmica

En aquesta secció es detallarà el cost que ha suposat dur a terme aquest projecte. Per a calcular els costos del projecte es consideraran els següents recursos:

- Recursos humans
- Recursos Software i Hardware

10.2.1. Recursos humans

Per a calcular el cost dels recursos humans s'hauran de considerar el total d'hores dedicades al projecte. Aquestes hores les subdividirem per assignar-les a cadascun dels rols que intervenen en aquest projecte software.

Aquest projecte ha estat desenvolupat per una única persona que ha assumit gairebé tots els rols que existeixen en el desenvolupament d'un projecte software. No obstant, s'indicarà el tipus de tasca i rol teòric associat a mode d'obtenir el cost total derivat dels recursos humans de la forma més precisa possible.

Els rols amb que dividirem les tasques estan basats en els rols detectats a l'inici del document, concretament a la secció 'Gestió del projecte', apartat 'Recursos', subapartat 'Recursos requerits', on es feia una pinzellada als tipus de recursos que un projecte d'aquestes característiques requereix. Val a dir, però, que només es consideraran aquells rols que afecten al desenvolupament del projecte de forma directa; altres rols, com per exemple el grup d'experts, es aliè al mateix i per tant no es considera.

Els rols considerats són els següents:

- Cap de projectes: Es tracta de la persona a qui recau la responsabilitat final del projecte. S'encarrega de les comunicacions amb el client, de seguir l'evolució del projecte i d'iniciar i finalitzar el mateix. També s'encarrega de realitzar els diferents documents a entregar.
- Analista/Arquitecte: S'encarrega d'analitzar els requisits del projecte i defineix com es realitzarà el desenvolupament del mateix.
- Programador: Persona que s'encarrega de la implementació del sistema. Generalment també s'encarrega de documentar aquelles funcions que implementa.

- Tester: Aquest rol el duen a terme les persones encarregades de comprovar i verificar cadascuna de les funcionalitats del sistema.

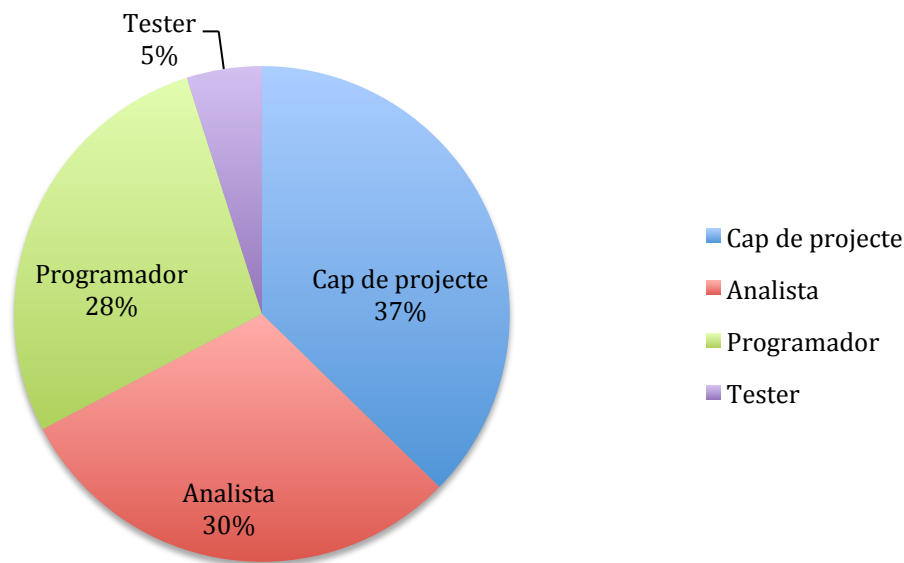
Amb els rols definits, es necessari calcular les hores de treball invertides en total. S'ha de considerar, però, que tot el temps que ha durat el projecte no ha sigut ni molt menys a temps complet. Només els mesos de setembre a desembre van contar amb una dedicació parcial, els mesos anteriors, primer per estudis i després per vacances no van disposar de tot el temps que es consideraria desitjable.

La taula següent mostra com s'han repartit els rols cadascuna de les tasques realitzades, i el nombre d'hores que es correspon a cadascuna d'elles:

Rol	Responsabilitats	Hores	Preu/h	Cost (€)
Cap de projecte	Project management	50	40	200
	Tasques prèvies	15		6000
	Documentació	55		2200
Analista/ Arquitecte	Tasques prèvies	45	30	1350
	Anàlisi i disseny	130		3900
	Documentació	50		1500
Programador	Implementació	220	25	5500
	Documentació	30		750
Tester	Proves	50	20	1000
	Documentació	5		100
Total		650		22500

Taula 16. Costos en recursos humans

La quantitat total de diners estimats per al projecte és de 22.500€ pel que fa als recursos. Un desglossament més acurat ens permetrà fixar-nos en el repartiment de diners invertits:



Il·lustració 51. Repartiment recursos humans/cost

En aquest gràfic es pot apreciar que són el Cap de projecte i l'Analista els rols que s'enduen poc més d'una tercera part del pressuposts, posant de rellevància que aquest projecte ha requerit un esforç inicial important pel que fa a estudis previs, adaptació i anàlisis, remarcant així un cop més el caràcter atípic del projecte.

10.2.2. Recursos software i hardware

El cost d'un projecte també necessita d'un càlcul en quant a recursos tecnològics necessaris per al desenvolupament del projecte.

En aquest aspecte, el cost es bifurca en dos components: software i hardware.

Els recursos software representen tots aquells programes que s'han necessitat per a confeccionar el sistema. Durant el transcurs del projecte s'ha procurat usar software exclusivament lliure, tot i que aquest objectiu no s'ha complert en tots els casos. Concretament, el software usat queda reflectit a continuació:

Software	Cost (€)
Fraise (editor de text)	0
Office 2011 (mac)	140
Eclipse	0
Windows (licència via FiB)	0
Linux	0
Macintosh	30
XUL	0

Firefox	0
Java	0
Total	140

Taula 17. Cost software desglossat

D'aquesta manera, la major part dels costos venen donats pels components hardware. Concretament es tracta únicament d'un ordinador usat tant per al desenvolupament de l'aplicació com per a elaborar la documentació descrita amb anterioritat.

Recurs	Cost (€)
Software	140
Hardware	1300
Total	1440

Taula 18. Costos Hardware i Software

10.2.3. Cost total

Finalment, podem calcular els costos totals del projecte sumant cadascun dels costos obtinguts parcialment, tenint en consideració els recursos analitzats anteriorment.

En la següent taula es mostra el total dels costos:

Recurs	Cost (€)
Humà	22500
Software	140
Hardware	1300
Total	23940

Taula 19. Cost total del projecte

Com es pot observar, el cost total del projecte ha sigut de **23.940€**, d'on la major part (un 94%) es destina als recursos humans que s'han usat.

11. Conclusions

Després d'haver analitzat, especificat, dissenyat i implementat una interfície amb capacitat per controlar diferents dispositius domòtics, es pot fer una reflexió del treball realitzat per veure en quina mesura s'han assolit els objectius inicials del projecte.

11.1. Objectius assolits

En el segon apartat del document s'han mencionat els objectius d'aquest projecte, els quals de forma resumida definien crear una eina multiplataforma que permetés controlar diferents dispositius domòtics, amb un alt grau de flexibilitat entorn a les configuracions visuals, de manera que pogués ser usat amb relativa facilitat per gent amb diferents tipus de dificultats motrius o cognitives.

A aquestes alçades, es pot dir que s'han complert la gran majoria dels objectius establerts, concretats en forma de requisits funcionals i no funcionals. Tot i això, contratemps aliens al desenvolupament del projecte no han permès a temps de redacció d'aquest document que la interfície es connectés als dispositius domòtics, ja que el software de connexió no estava adaptat per a rebre peticions externes i no s'ha pogut adaptar a temps per la persona responsable d'aquell projecte.

A continuació, es repassaran els principals objectius que han quedat coberts:

11.1.1. Adaptació a les necessitats de l'usuari final

Es permet crear un nombre indeterminat de pantalles, on cada pantalla pot contenir un nombre elevat de botons, configurables en aparença i funcionalitat. Amb aquestes opcions, es permet adaptar el comportament de l'aplicació en base a les característiques de l'usuari final.

S'han considerat la capacitat de percebre l'entorn de l'usuari, així com la capacitat d'interacció amb la realitat que l'envolta alhora d'establir el sistema òptim per al major tipus de configuracions possibles.

11.1.2. Interacció amb dispositius domòtics

El sistema està preparat per connectar-se a un sistema DLA que governi els diferents dispositius domòtics de la casa. D'aquesta forma es pot associar una acció determinada a un dispositiu concret.

11.1.3. Configuració personalitzada

Les configuracions de l'aplicació son adients quan major és el grau de personalització de la interfície als gustos del consumidor, sempre respectant les bases que estableix el sistema per al seu funcionament. L'editor de recursos permet afegir tots aquells

Amb l'editor de recursos es poden afegir al sistema totes aquelles ajudes auditives i visuals que s'adaptin millor a l'usuari final; tanmateix, també es pot adaptar els colors de manera que resulti més senzill visualitzar certs elements de pantalla si es presenten deficiències visuals.

11.1.4. Vista prèvia

Es permet que prement només un botó, la persona encarrega de la configuració de la pantalla tingui una visualització completa de com es la configuració actual si es decidís fer definitiva. Això permet disminuir el temps requerit per a la l'adaptació del sistema al client, ja que es poden fer proves i visualitzar-les de forma ràpida i sense generar residus innecessaris en el sistema.

11.1.5. Independència de l'entorn de generació respecte al d'execució

Concebre un sistema amb un alt nivell d'usabilitat contempla que els usuaris que administren l'aplicació no es trobin limitats en quant a l'entorn d'execució del client final.

En aquest sentit, no només l'aplicació d'administració es presenta amb suport per als tres principals Sistemes Operatius d'escriptori, sinó que s'ha dotat al mateix amb la capacitat de generar l'aplicació des de qualsevol Sistema Operatiu a qualsevol altre diferent.

11.2. Conclusions del projecte

La realització d'aquest projecte, es a dir, la creació d'una interfície gràfica que s'ha de comunicar amb diferents dispositius domòtics, ofereix el valor afegit respecte altres solucions en el fet que ofereix mètodes d'adaptació de la interfície gràfica pensats per adaptar-se a les dificultats cognitives o motrius de l'usuari final; intentant millorar la qualitat de vida d'aquests usuaris.

A nivell tècnic, cal remarcar que aquest projecte treballa amb un gran nombre de tecnologies diferents:

- El motor XUL ha facilitat la creació d'una interfície gràfica robusta i rica en elements; es fàcil d'usar i ha permès crear una interfície multiplataforma amb cost pràcticament 0.
- La persistència de les dades s'ha realitzat usant el motor Javascript, juntament amb mòduls XPCOM del motor XUL, prescindint de les bases de dades i gestionant de manera directa les dades a emmagatzemar.
- Java es un motor potent, multiplataforma i altament provat que ha afegit les eines necessàries per comunicar-se de forma transparent amb el servidor DLA i, de retruc, amb els dispositius domòtics existents.

Cal també considerar que aquest projecte està emmarcat dins d'un prisma més gran que el purament tècnic, i es que al integrar la interfície gràfica adaptable amb la interfície que aglutina tots els dispositius domòtics d'una casa o centre en un sol punt, s'està aconseguint una solució final que no es trobava disponible dins de l'àmbit de la promoció de l'autonomia personal per a persones amb algun tipus d'incapacitat. Doncs s'aconsegueix un sistema que no depèn ni dels sistemes domòtics que el componen (es prescindeix dels múltiples sistemes de control creats pels fabricants), ni del sistema operatiu que l'ha d'executar; i que alhora aquest sistema ha estat adaptat per facilitar l'accés a aquestes tecnologies domòtiques.

Aquest projecte ha estat orientat dins del marc de col·laboració entre la FIB i el CVI. De fet, es molt interessant saber que el projecte que s'ha realitzat està orientat amb la finalitat de ser usat en el centre CVI, amb perspectives a posar-ho en funcionament tant aviat com sigui possible. Desenvolupar un sistema amb la certesa que correspon a una necessitat real, i que per tant tindrà un ús real, suposa també un al·licient durant l'etapa de desenvolupament.

11.3. Valoració personal

Com a valoració personal un cop finalitat el projecte, després de dedicar moltes hores de treball, solucionar problemes, reestructurar alguns aspectes del mateix.. però sobretot veure el resultat obtingut i l'ús que se li pot arribar a fer, puc dir que ha sigut una experiència molt gratificant.

Primer, pel fet de tractar-se d'un projecte d'una gran envergadura, en el que he hagut d'adoptar diferents rols durant el mateix: Cap de projectes, Analista, Dissenyador, Programador, Tester... Durant el període a la FIB ja havia realitzat projectes on havia fet algun d'aquests rols, però en cap cas requereien totes les responsabilitats sota una mateixa persona, ja que es repartien els rols i s'especialitzava cadascú en la part que li corresponia. Aquest cop els he hagut de fer tot jo.

En segon terme, aquest projecte ha suposat per mi una motivació personal extra. Durant el transcurs de finalització de la carrera i cerca de PFC, vaig estar mirant alguns dels projectes que oferia la universitat i també vaig parlar amb algun dels professors que més bona impressió em van deixar durant els estudis, però als projectes que trobava els hi faltava, al meu parer, un plus o motivació extra. Sens dubte es tractava de projectes ben vàlids per acabar la carrera, però no considerava que aportessin res a nivell personal: la majoria de projectes estan enfocats o bé a nivell intern o bé a consum a nivell empresarial. Aquest projecte, però, pretén de forma activa ajudar en la vida dels demès, ajudar a un col·lectiu de gent del que no només algun dia tots nosaltres ens hi veurem abocats, sinó que tradicionalment i socialment s'ha trobat més desplaçat de la resta de persones.

Finalment, també ha suposat un repte a nivell personal responsabilitzar-me d'un projecte amb unes característiques que no havia trobat a la carrera, doncs no només he hagut de treballar amb tecnologia inicialment nova i desconeguda, sinó que aquest projecte aportava una sèrie de requeriments no funcionals que superava el que per norma es contempla a la carrera: era més prioritari satisfer les necessitats d'usabilitat i accés que no les de funcionalitat. Sovint tendim a pensar que un projecte es bò si tècnicament aplica tots aquells conceptes que t'han ensenyat, si els mètodes aplicats son eficients, si el resultat final es prou complicat com per a auto-justificar-se; però aquests últims anys he estat canviant una mica aquesta mentalitat, considerant que un projecte ha de ser simple, fàcil d'usar i ha de centrar-se en els usuaris finals i de quina manera pot ajudar el que es vol implementar a millorar la situació dels usuaris finals.

Per últim, no voldria acabar aquest projecte sense agrair de forma pública al meu tutor Cristian Barrué per la oportunitat que se m'ha donat en aquest projecte, i el

temps que ha invertit en mi. Així com a la meva família, amics i especialment la meva parella, per la paciència que han tingut i el suport que m'han donat durant tot aquest temps. Gracies a tots!

12. Propostes de millora

Encara que els objectius principals han quedat coberts, tot projecte sempre passa per unes etapes en les que el personal implicat en el projecte veu millores possibles, o alternatives que originalment no s'havien considerat i que podrien fer un gran servei al projecte, però que per temes de calendari, personal o pressupost, no es viable realitzar-los durant la realització original del projecte.

Algunes de les propostes que he considerat més interessant són:

12.1. Comandes de veu

La interfície gràfica pretén ajudar al nombre màxim de persones, aquesta es una realitat que ja s'ha anat comentat al llarg del projecte. Per ajudar a un col·lectiu dels usuaris finals, es disposa d'eines que permeten reproduir so, a fi de reforçar per mètodes auditius les funcionalitats que ha realitzar l'aplicació.

El mètode emprat, permet llegir fitxers de so i reproduir-los, de forma que qualsevol Sistema Operatiu en pot fer us d'ells i permet aportar una veu personal i pròxim a l'usuari final, si es considera necessari.

Tot i això, usar mètodes de reconeixement de text aportaria al sistema una flexibilitat superior al sistema. No es tracta d'eliminar les funcionalitats de so ja existents, sinó de buscar un mètode alternatiu que s'adapti a totes les plataformes i permeti llegir automàticament el text associat a objectes.

12.2. Detector de presència – generació automàtica de pantalles

L'aplicació actual, tal i com està plantejada, requereix que el sistema es configuri prèviament mitjançant una interfície d'administració. Amb aquesta interfície gràfica es poden situar els diferents elements de la casa de la forma que es consideri més adient.

Ara bé, aquesta feina es podria automatitzar molt més si es disposes d'elements capaços de detectar la presència. En aquest cas, es podria dissenyar una interfície que, fent ús dels elements clau de la interfície gràfica actual, fos capaç de detectar la sala actual, demanes al DLA els dispositius corresponents a la sala i s'autoconfigures amb els dispositius existents en aquella sala.

Aquest mètode, però, requeriria un anàlisi en més profunditat, per veure de quines maneres podrien coexistir els mètodes de configuració manuals i

automàtics. També existiria la problemàtica sobre quins criteris usar alhora de crear la pantalla de forma dinàmica.

12.3. Augment de formats suportats

Una de les últimes propostes de millora té molt a veure amb la tecnologia emprada per crear les interfícies gràfiques: XUL.

Aquesta tecnologia és multiplataforma i compleix molt bé les seves funcionalitats, però presenta un handicap en quant als formats de fitxers que suporta, ja que està basada en tecnologies completament lliures i, per tant, totes aquelles tecnologies que requereixen llicència (com el MP3 o MP4) no tenen suport oficial.

Tot i que no afecta a la funcionalitat del programa, aquesta millora aporta un extra a la comoditat que ha de suposar treballar amb la interfície gràfica, ja que el responsable de la seva configuració no ha de preocupar-se dels formats dels fitxers.

Un mètode proposat per millorar aquests sistema consisteix en implementar funcionalitats de reproducció amb Java o C++. El problema principal, es que aleshores la interfície seria dependent d'aquesta tecnologia i s'hauria de tenir en consideració de cara a futures actualitzacions o portabilitats.

Actualment, fer que la interfície depengués de Java no seria cap inconvenient, ja que la connexió amb el DLA està feta en Java i per tant ja existeix aquest requeriment tecnològic, però si es canvia el mètode de connexió als dispositius domòtics en un hipotètic futur, s'hauria de considerar que la interfície continuaria essent dependent de Java.

13. Bibliografia

Informació sobre l'envelliment de la població:

- Informe de l'ONU, efectuat el 2009.
 - http://www.un.org/esa/population/publications/WPA2009/WPA2009_WorkingPaper.pdf
- Resum amb conclusions sobre l'informe.
 - http://es.wikipedia.org/wiki/Envejecimiento_de_la_población

Documentació sobre discapacitats motrius i cognitives

- Consideracions en el disseny de la interfície gràfica: document del projecte **Interaction Design Report and Middleware**
 - <http://www.ist-shareit.eu/shareit>

Anàlisi de d'interfícies hardware:

- Tipus de dispositius apuntadors.
 - http://en.wikipedia.org/wiki/Pointing_device

Centre de Desenvolupament per a la Independència:

- Pàgina oficial del centre
 - <http://www.cvi-bcn.org/>

Tecnologia DLA

- Tecnologia extreta de la Tesi Doctoral realitzada per Eduardo Javier Pérez Rodríguez en la Universitat de Màlaga, sota el títol "Arquitectura de navegación distribuïda para agentes robóticos".
 - <http://dialnet.unirioja.es/servlet/tesis?codigo=18278>

Pàgina oficial del XUL: <http://developer.mozilla.org/es/XUL>.

Desenvolupament XUL bàsic:

- Introducció al desenvolupament en XUL:
 - <http://brian.kingsonline.net/talk/projects/writings/xulrunner-application-development/>
- "**Developing GUIs with XUL**", per Eitan Suez
 - <http://www.uptodata.com>
- Implementació i integració d'una aplicació XUL al Sistema Operatiu.
 - https://developer.mozilla.org/en/XULRunner/Deploying_XULRunner_1.8
 - <http://brian.kingsonline.net/talk/projects/writings/xulrunner-application-development/>

Tutorials XUL:

- Pàgina oficial amb exemples que descriuen cadascuna de les diferents parts de la interfície d'usuari.
 - https://developer.mozilla.org/en/XUL_Tutorial
- Mètodes bàsics per a la creació d'una interfície XUL. També s'inclouen comparacions entre HTML5 i XUL i com crear l'estructura de fitxers.
 - <http://www.xul.fr/tutorial/>

Crida a codi Java des de XUL:

- Cridar Java des d'aplicacions XUL.
 - <http://stanislavvitvitskiy.blogspot.com/2009/04/calling-java-from-xul-applications.html>
- Suport a LiveConnect en la nova tecnologia de plug-ins Java.
 - <https://jdk6.dev.java.net/plugin2/liveconnect/>
- Grup de discussió sobre la invocació de mètodes Java des de XUL.
 - http://groups.google.com/group/mozilla.dev.tech.xul/browse_thread/thread/0adef62862f3a827/1a5845274b2d5781
- Cridant Java des de Javascript.
 - <http://www.andonyar.com/rec/2008-12/gemse/demo/doc/dev/java.xhtml>

Aplicacions de referència usades durant el període d'anàlisi de la tecnologia:

- Gemse: Editor de fórmules matemàtiques.
 - <http://www.andonyar.com/rec/2008-12/gemse/approach1>
- Llistat d'aplicacions basades en la tecnologia XULRunner. Actualment es troba una mica desfasat, però serveix per obtenir un llistat de programes de referència i posterior estudi.
 - https://developer.mozilla.org/en/XULRunner_Hall_of_Fame
- XULPlayer: reproductor multimèdia per a sistemes Windows.
 - <http://xulplayer.sourceforge.net/>

Recerca desenvolupament components XPCOM:

- Creació d'elements XPCOM: "**Creating XPCOM Components**", per Doug Turner i Ian Oeschger.
- "**Scripting Mozilla Applications with XPCOM and XUL**", d'Andreas Mitschek com a tesis del curs "IT-Spezialisierung E-Commerce Vertiefungskurs VI WS 2008/09".

14. Annexos

14.1. Mètodes d'interacció amb una interfície gràfica

En el transcurs del projecte s'ha necessitat investigar sobre els diferents mètodes d'interacció existents en el centre CVI i d'altres mètodes comuns que permetin al client final interactuar amb el sistema software. Fruit d'aquestes investigacions, s'ha creat aquest document, que pretén aglutinar tots aquells mètodes que s'han considerat alhora de dissenyar la interfície gràfica.

Aquests mètodes es poden classificar segons el mètode físic d'interacció i qui es responsabilitza del seu control:

Hardware

Es refereix a tots els dispositius físics que s'usen per a la interacció amb el software. Entre els objectes d'estudi es troben:

- Ratolí.



Il·lustració 52. Interfície hardware: ratolí

És la interfície hardware de control d'interfícies per excel·lència. Utilitza la mà com a mètode de control i sovint consta de (mínim) dos botons, apart d'un detector de moviment que es sincronitza amb el punter que es situa a la pantalla del ordinador.

- Joystick



Il·lustració 53. Interfície hardware: joystick tradicional

Aquesta interfície es més pròpia del món del videojoc, tot i que actualment s'usen joysticks per al control de robots, llits amb motor elèctric.. en general, tots aquells dispositius hardware en els que pugui controlar el seu moviment. Incorpora una palanca per definir els moviments (també pot ser una creueta amb els quatre eixos direccionals) i, normalment, un o més botons adaptables per definir accions.

A nivell global aquests dispositius s'usen en el centre CVI, però no s'han considerat elements clau en la interacció de la interfície gràfica ja que no tenen un ús extens en el Sistemes Operatius fora del camp dels videojocs.

- Polsador



Il·lustració 54. Interfície hardware: polsador

Un polsador seria un ratolí al que se li treu el hardware corresponent a la detecció de moviment i se'l deixa només amb un botó. Aquest tipus d'interfícies hardware són comuns trobar-les en ús per gent que té problemes motrius que afecten a la capacitat de mantenir els músculs relaxats o tensos de forma voluntària; es tractaria de persones que pateixen, per causes diverses, d'espasmes musculars.

Aquesta interfície hardware ha de permetre la interacció dels usuaris amb la interfície gràfica dissenyada pel projecte. Per tant, va caldre dissenyar un mètode d'interacció software que afavorís aquest tipus de hardware, tal i com queda reflectit en el següent apartat, on s'expliquen les interaccions gràfiques a nivell de software, concretament, el mètode d'escaneig.

- Detectores de moviment ocular



Il·lustració 55. Interfície hardware: detector de moviment ocular

Aquest és un mètode innovador, relativament poc usat, que permet interactuar amb un Sistema Operatiu només mirant la secció de la pantalla a la que es vol anar. Consisteix d'un detector de moviment especialitzat en detectar els ulls i seguir el moviment de la pupil·la; després, només queda traduir els moviments de la pupil·la en els moviments que faria un ratolí tradicional.

Com a mètode d'interacció respecte la interfície gràfica desenvolupada en el projecte, no ha suposat cap canvi o innovació al respecte.

- Apuntador



Il·lustració 56. Interfície hardware: apuntador

S'usa en les pantalles tàctils, especialment en les de primera generació. Es basa en un punter, semblant a un bolígraf, per substituir les funcions del ratolí tradicional. Es basa en la pressió en comptes del moviment per definir l'àrea de la pantalla que s'està usant.

- Pantalla tàtil



Il·lustració 57. Interfície hardware: pantalla tàtil

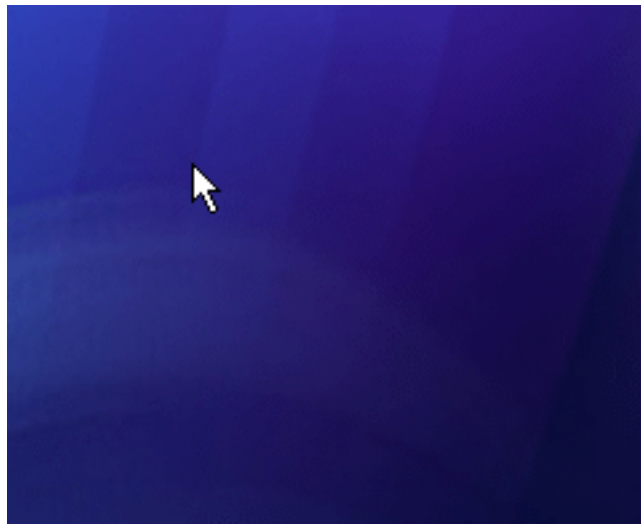
És la evolució lògica dels sistemes tàctils basats en apuntadors. Substitueixen els apuntadors per el dit humà, permetent una interacció més directa i transparent entre el sistema software i la persona encarregada d'usar el sistema.

A nivell del projecte, té certes implicacions que no aporta un sistema tàctil basat en apuntadors: la mida mínima dels objectes. S'ha de considerar que un dit humà es substancialment més gruixut que la punta d'un apuntador, fet que s'ha de considerar al dissenyar la interfície gràfica, doncs no s'han de dissenyar objectes que l'usuari final no pugui clicar per que siguin massa petits.

Software

Es fa referència als mètodes de control d'interfície que s'han considerat a nivell de software per a la interacció de la interfície gràfica amb l'usuari final:

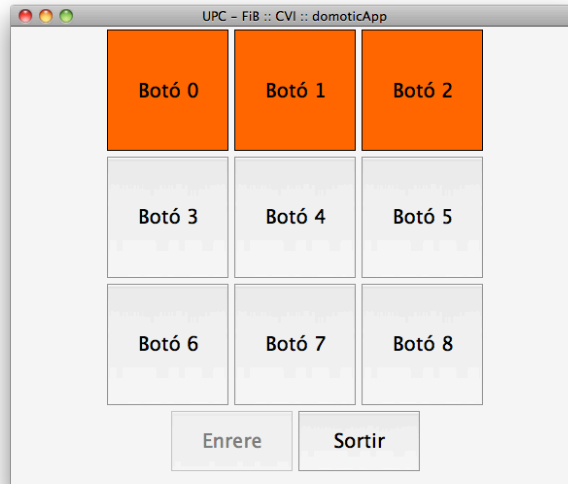
- Punter.



Il·lustració 58. Interfície software: punter

És el mètode clàssica d'interacció de qualsevol Sistema Operatiu d'escriptori amb els elements que es representen a la pantalla. Mitjançant algun tipus d'interfície hardware es correlaciona la posició del punter amb l'objecte que es vol seleccionar.

- Escaneig.



Il·lustració 59. Interfície hardware: escaneig

Aquest mètode és propi de la interfície gràfica desenvolupada per aquest projecte, i ha sorgit com a resposta a la necessitat d'usar interfícies hardware com els polsadors.

El seu principi bàsic de funcionament és recórrer de forma automàtica tots els elements de la interfície, a la espera que l'usuari final enviï algun tipus de senyal que indiqui que vol seleccionar l'objecte actual (generalment es tractarà d'interfícies hardware tipus polsador, tot i que accepta qualsevol interfície que emuli el clic d'un ratolí). Aquest mètode d'escaneig es pot configurar segons les quatre vies d'escaneig clàssiques: començant per la dreta, esquerra, amunt o avall. També inclou la possibilitat d'efectuar escaneig lineal, de forma que es pugui accedir amb més rapidesa a l'element objectiu.

L'escaneig lineal tracta de recórrer els objectes en bloc, línia a línia, fins que l'usuari arriba al bloc que conté la funcionalitat desitjada. Al prémer un botó, es selecciona el bloc i es passa a fer un escaneig element a element dins d'aquell bloc, permetent a l'usuari seleccionar finalment l'element desitjat. És fàcil apreciar que amb aquest sistema es pot reduir considerablement el temps de cerca requerit per arribar a una funcionalitat concreta.

14.2. Manual d'instal·lació d'entorn

En aquest manual s'explicarà com instal·lar i configurar l'aplicació XUL; també s'explicarà com s'estructura una aplicació XUL genèrica i quines diferències principals hi ha entre les versions Windows, Linux i Macintosh.

Primerament, abans d'iniciar l'aplicació és necessari tenir instal·lat alguna màquina virtual de Java. Si ja es disposa d'una màquina virtual de Java, es pot ometre aquest pas; altrament, es pot descarregar la màquina virtual a <http://www.java.com/es/download/> per instal·lar-la.

El següent pas es iniciar el servidor DLA, que detecta els dispositius existents i permetrà a la interfície comunicar-se amb ells.

Si es disposa d'un sistema Windows, s'haurà d'executar la següent comanda:

```
java DLAServer
```

En cas de Linux o Macintosh, s'ha d'obrir un terminal, accedir a la carpeta amb el servidor i executar la següent comanda:

```
./DLAServer
```

Amb el servidor en marxa, ja podrem executar l'aplicació. Primerament s'ha de descomprimir l'aplicació, generant una carpeta amb el nom CVI. Cal recordar que abans d'iniciar l'aplicació, cal tenir la part del servidor funcionant, tal i com s'ha explicat anteriorment.

Al executar el servidor, aquests ens retornarà les dades d'IP i Port amb les que s'haurà d'actualitzar el fitxer "dla.ini", situat a la carpeta Java del sistema. Aquest fitxer indica a la interfície gràfica com es pot comunicar amb el DLA.

Configurat el fitxer de connexió amb el DLA, ja es pot executar l'aplicació:

- Si es disposa de la versió Windows, s'haurà d'executar la comanda: CVI . exe
- En cas de funcionar sota Linux, executar la comanda: ./CVI, o fer doble clic sobre l'icona si el sistema ho permet.
- Si es tracta d'un Macintosh, només s'haurà de fer doble clic a la icona amb nom CVI.

14.3. Manual d'ús

Aquest annex està dedicat a oferir un manual d'ús de la interfície gràfica

A l'apartat 7 (Disseny gràfic) ja es pot observar una descripció de la interfície gràfica, fet que ajuda en gran mesura a entendre la distribució dels elements en pantalla. Per aquest motiu, en aquest manual d'ús es partirà de la base que el lector ja ha llegit prèviament aquest apartat, i per tant està familiaritzat amb els diferents elements d'interacció de la interfície gràfica. D'aquesta forma, el manual es centrarà exclusivament amb el seu ús.

Tal i com s'ha comentat a l'apartat 7, la interfície gràfica es divideix segons el tipus d'usuari: Administrador o client.

Interfície d'administrador

L'usuari de tipus Administrador té com objectiu configurar la interfície gràfica de forma que s'adapti a les necessitats de l'usuari final.

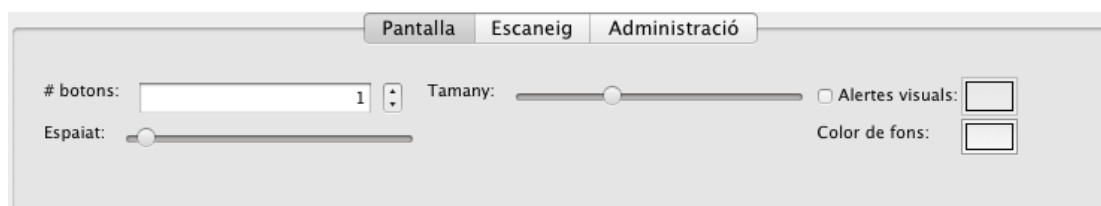
A continuació es mostren les diferents funcionalitats que ofereix la interfície:

Pantalla

Aquesta secció presenta totes les opcions que es poden realitzar en la interfície gràfica i que afecta aspectes visuals concrets a la pantalla que s'està visualitzant. Totes les opcions de pantalla afecten només a la pantalla actual, permetent, per tant, configuracions diferents segons la pantalla actual.

Per accedir a les opcions d'edició de la pantalla, s'ha de clicar la pestanya "Pantalla", situada a la part superior de la interfície gràfica.

La representació en pantalla dels botons es genera de forma automàtica al fer qualsevol canvi en els atributs dels mateixos.



Il·lustració 60. Opcions de pantalla

Afegir/Eliminar botons

Per afegir o eliminar un botó, ens situarem a la opció superior esquerra del requadre.

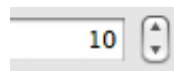
Des d'aquí, es pot modificar el nombre de botons de dues formes:

- Introduint el nombre de botons que es desitgen al requadre de text.

A screenshot of a user interface element. It consists of a light gray rectangular box. On the left side of the box, the text "# botons:" is displayed in a dark font. To the right of this text is a white rectangular input field with a thin gray border. Inside the input field, the number "9" is centered. To the right of the input field, there is a small gray square containing the number "9".

Il·lustració 61. Introducció manual # botons.

- Usant les funcions d'afegir i eliminar contigües al requadre de text. Amb aquest mètode s'afegeix o eliminen botons d'un en un.

A screenshot of a user interface element. It consists of a light gray rectangular box. On the left side of the box, there is a white rectangular input field with a thin gray border, containing the number "10". To the right of the input field is a vertical spin button with a white background and a gray border, featuring a small upward-pointing triangle at the top and a small downward-pointing triangle at the bottom.

Il·lustració 62. Funcions semi automàtics per al # botons

Mida dels botons

Per modificar la mida d'un botó, ens situarem al centre del requadre amb les opcions de pantalla.

Al moure la barra cap a la dreta o cap a l'esquerra, es modifica la mida de les icones de la pantalla actual.

A screenshot of a user interface element. It consists of a light gray rectangular box. On the left side of the box, the text "Mida:" is displayed in a dark font. To the right of this text is a horizontal slider bar with a thin gray border. A blue circular knob is positioned in the center of the slider bar.

Il·lustració 63. Barra d'ajustament de la mida

Espaiat dels botons

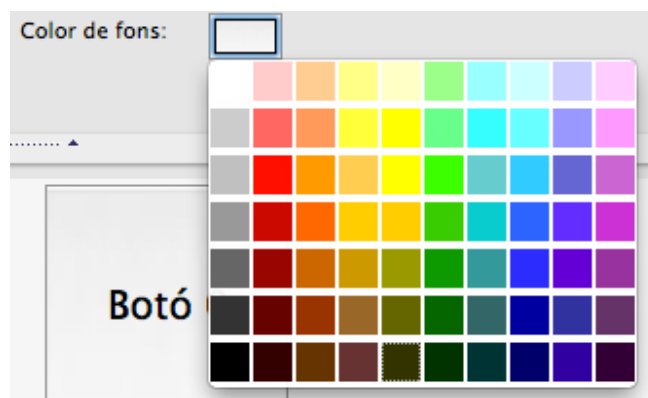
Aquesta opció es troba a l'extrem esquerra inferior del requadre d'opcions. Es tracta d'una barra amb valors màxims i mínims predefinitos; al moure aquesta barra cap a la dreta o cap a l'esquerra, es calcula el marge entre els botons.

A screenshot of a user interface element. It consists of a light gray rectangular box. On the left side of the box, the text "Espaiat:" is displayed in a dark font. To the right of this text is a horizontal slider bar with a thin gray border. A blue circular knob is positioned on the left side of the slider bar.

Il·lustració 64. Barra d'ajustament de l'espaiat entre botons

Color de fons de l'aplicació

Per canviar el color de fons de la pantalla actual, cal situar-se a l'extrem inferior dret de les opcions de pantalla. Amb el text "Color de fons", es troba un botó, que al clicar-lo desplega una paleta de colors. Seleccionant qualsevol dels colors es modifica el color de fons actual per el nou escollit.



Il·lustració 65. Modificar color de fons de l'aplicació

Activar/desactivar ressaltat dels botons

De cara a ressaltar els botons, primer cal decidir si es vol activar aquesta opció. El ressaltat és una opció important ja que si s'activa, el color escollit per al ressaltat serà el que s'utilitzi en les opcions d'escaneig, si es que aquestes també s'usen en la interfície d'usuari.

Per activar-la o desactivar-la, ens situarem a l'extrem superior dret del requadre amb les opcions de pantalla, on es troba una casella al costat dret del text "Alertes visuals", si es clica un cop, s'activen les opcions d>alertes visuals i el botó amb els color passa a ser clicable; si es desactiva la casella, el botó de selecció de colors es desactiva i no es pot clicar.

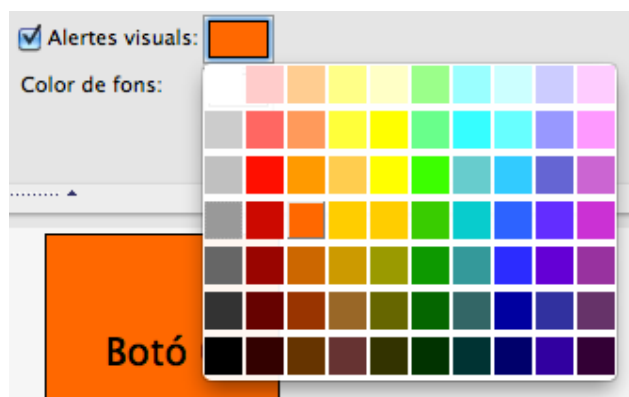


Il·lustració 66. Activar alertes visuals

Color de ressaltat de botons

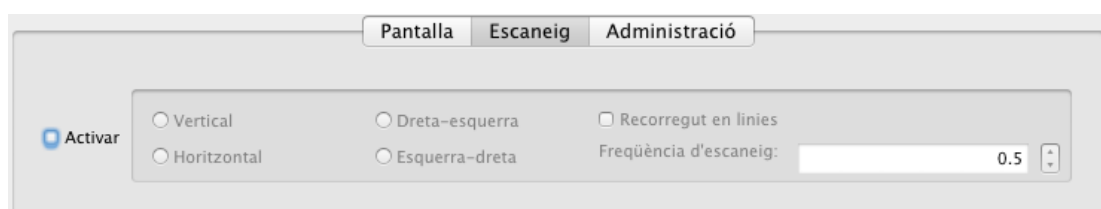
Aquesta funcionalitat requereix que prèviament s'hagi activat la casella corresponent al ressaltat de botons, tal i com explica el punt que precedeix l'actual.

Amb la casella activa, només cal prémer el botó que hi ha al costat dret del text "Alertes visuals" i es desplegarà un requadre amb una paleta de colors. Al seleccionar un color aquest quedarà marcat per al ressaltat, i només quedarà passar per sobre d'un botó per comprovar que el ressaltat s'ajusta a l'esperat.



Il·lustració 67. Ressaltat de botons mitjançant colors

Escaneig

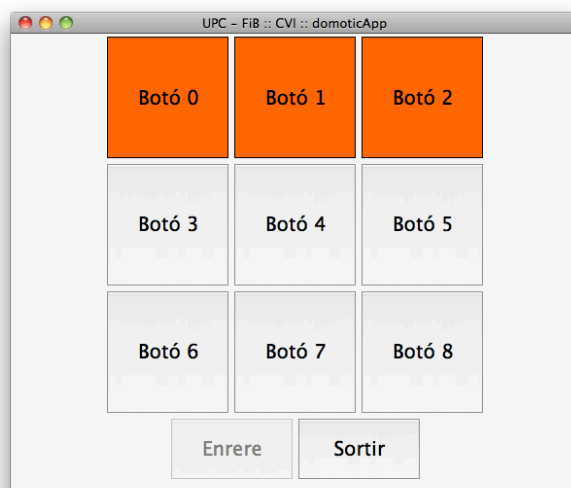


Il·lustració 68. Opcions d'escaneig

Aquesta secció presenta totes les opcions que es poden realitzar en la interfície gràfica i que afecta als aspectes de navegació automàtica entre els diferents elements visuals concrets. Les opcions de configuració d'escaneig afecten a totes les pantalles del sistema; per tant s'haurà de configurar considerant cadascuna de les diferents pantalles. La raó és que al tractar-se d'un mètode per facilitar l'accés, s'ha considerat que modificar el comportament pantalla a pantalla repercutiria negativament amb la percepció de la utilitat, podent descentrar al usuari final.

Per accedir a les opcions d'escaneig de la pantalla, s'ha de clicar la pestanya "Escaneig", situada a la part superior de la interfície gràfica.

La representació en pantalla de les opcions d'escaneig no es visualitza en pantalla ja que entraria en conflicte amb la configuració de botons, i alhora seria un element de distracció per a l'administrador del sistema. Tot i això, sí que entra en funcionament tant al fer una "Vista prèvia" com al generar l'aplicació client.



Il·lustració 69. Opcions d'escaneig en funcionament

Activar escaneig

Es tracta de la casella responsable de permetre l'edició de la resta de paràmetres de l'escaneig. Es troba a l'extrem esquerra del requadre d'opcions d'escaneig. Al activar-se, permet modificar la resta d'elements d'escaneig, i al desactivar-se, inhabilita les opcions de configuració d'escaneig i fa que l'escaneig no s'activi tant al fer una 'Vista prèvia' com al generar l'aplicació final.



Il·lustració 70. Activar opcions d'escaneig

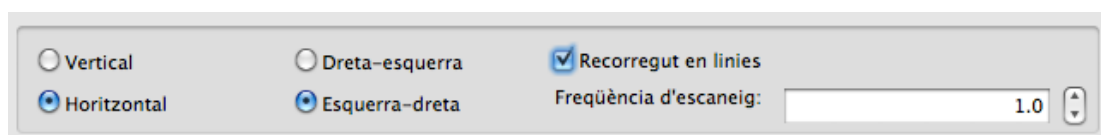
Opcions d'escaneig

Un cop activa la casella d'escaneig (veure punt anterior), s'activen les opcions d'escaneig.

Des del panell d'escaneig es pot indicar al sistema si es desitja fer un escaneig de tipus vertical o horitzontal i si es desitja fer-lo en sentit dreta-esquerra o

viceversa. Cal remarcar que l'escaneig no es pot activar si no es tenen mínim aquestes dues opcions definides.

També se li pot indicar l'escaneig lineal (veure Annex "Mètodes d'interacció amb una interfície gràfica" per a més informació), així com la freqüència d'escaneig, que es el nombre de segons que s'ha d'esperar la interfície gràfica abans de canviar d'element o conjunt d'elements seleccionats.

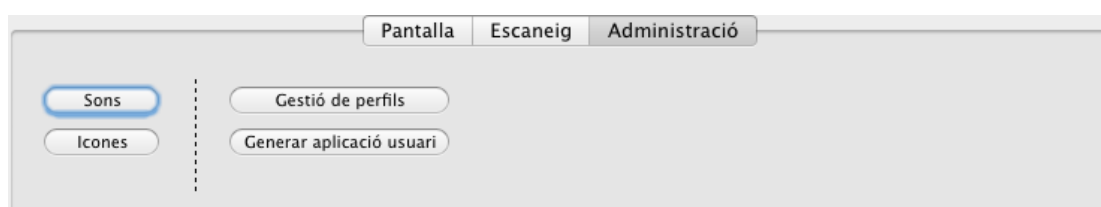


Il·lustració 71. Opcions d'escaneig

Administració

Aquesta secció presenta totes les opcions que es poden realitzar en la interfície gràfica i que afecta als aspectes de configuració de la interfície i els recursos de que disposa. Des d'aquesta pantalla se li dirà a la interfície quines imatges o sons disposa, així com també guardar/carregar configuracions prèvies i generar l'aplicació que l'usuari final haurà d'usar.

Per accedir a les opcions d'administració, s'ha de clicar la pestanya "Administració", situada a la part superior de la interfície gràfica.



Il·lustració 72. Opcions d'administració

Afegir sons

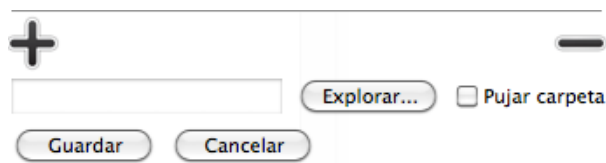
Per afegir un so, primer s'ha de clicar el boto situat a l'extrem superior esquerra del panell d'administració.

Al clicar-lo es desplegarà el següent quadre:



Il·lustració 73. Panell de gestió de sons

Si es clica al boto “+”, situat a l’extrem inferior esquerra del panell, apareixeran les opcions per afegir un nou so:



Il·lustració 74. Opcions per afegir sons

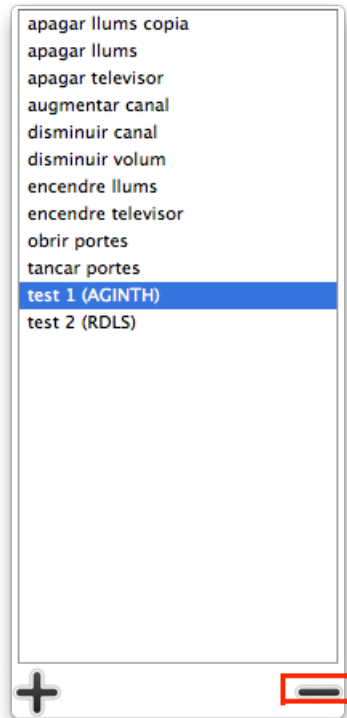
Si es clica el botó “explorar” es podrà navegar pel sistema fins a trobar el so corresponent. Cal remarcar que els formats de sons suportats pel sistema son els arxius de tipus WAV i de tipus OGG.

Un cop seleccionat el fitxer es retornarà a la pantalla amb el panell desplegat i l’adreça marcada. Si es clica la pestanya “Pujar carpeta”, es pujaran tots els elements corresponents a sons que es trobin a la carpeta indicada. Finalment, al prémer el botó “Guardar”, es copiarà el fitxer al sistema i ja es podrà usar en la configuració dels botons.

Eliminar sons

Per eliminar un so primerament s'ha de carregar el panell de sons (veure punt anterior, "Afegeir sons", per saber com accedir al panell).

Amb el panell desplegat, es selecciona del llistat aquell so que es vulgui eliminar i tot seguit es prem el boto "-" situat a l'extrem inferior dret del panell.

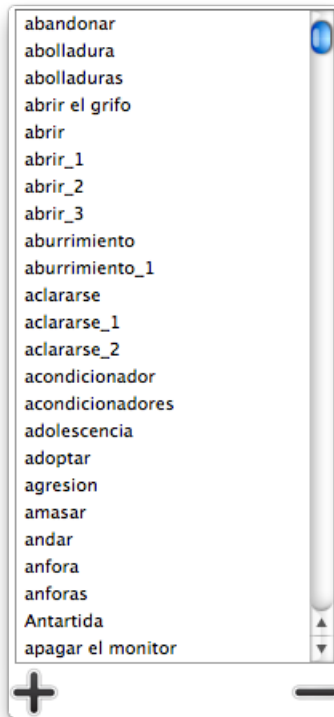


Il·lustració 75. Eliminar sons

Afegeir imatges

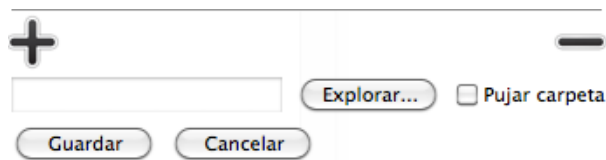
Per afegir una imatge, primer s'ha de clicar el boto situat a l'extrem inferior esquerra del panell d'administració, titulat "Icones".

Al clicar-lo es desplegarà el següent quadre:



Il·lustració 76. Panell imatges disponibles en el sistema

Si es clica al boto “+”, situat a l’extrem inferior esquerra del panell, apareixeran les opcions per afegir una nova icona:



Il·lustració 77. Opcions per afegir imatges

Si es clica el botó “explorar” es podrà navegar pel sistema fins a trobar la imatge corresponent

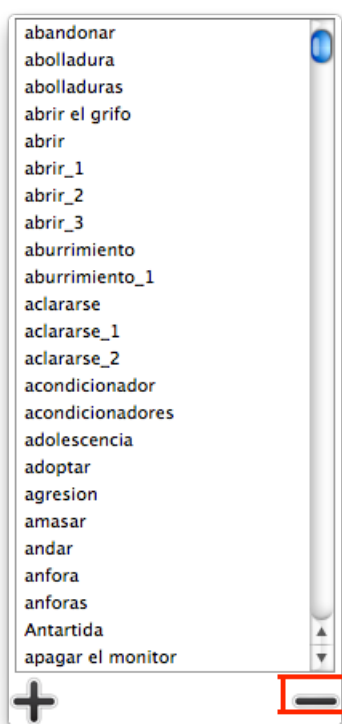
Un cop seleccionat el fitxer es retornarà a la pantalla amb el panell desplegat i l’adreça marcada. Si es clica la pestanya “Pujar carpeta”, es pujaran tots els elements corresponents a imatges que es trobin a la carpeta indicada. Finalment, al prémer el botó “Guardar”, es copiarà el fitxer al sistema i ja es podrà usar en la configuració dels botons.

Eliminar imatges

Per eliminar una imatge, primer s'ha de clicar el boto situat a l'extrem inferior esquerra del panell d'administració, titulat "Icones".

Al clicar-lo es desplegarà un panell, com es pot observar en l'apartat "Afegeir imatges", il·lustració 68.

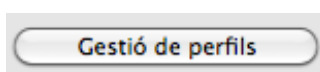
Es selecciona la imatge que es desitgi eliminar i es prem el boto "-", situat a l'extrem inferior dret del panell. Al prémer el botó s'eliminarà automàticament la imatge seleccionada.



Il·lustració 78. Eliminar imatge del sistema

Accedir als perfils

Per accedir als perfils emmagatzemats, s'ha de clicar el botó situat a l'extrem superior dret del panell d'administració, tal i com s'explica en la introducció d'aquest mateix apartat.



Il·lustració 79. Accés a la gestió de perfils

Al prémer sobre el botó, apareixerà un panell que conté el llistat de perfils actualment disponibles en el sistema, així com les opcions pertinents associades a perfils.



Il·lustració 80. Panell amb els perfils disponibles en el sistema

Guardar perfil

Primerament, cal accedir als perfils, tal i com s'explica en l'entrada "Accedir als perfils".

Un cop es disposa del panell amb la informació relativa als perfils, per guardar el perfil actual cal, primerament, prémer el botó amb un "+".



Il·lustració 81. Botó per afegir perfils

Al prémer el botó es desplegarà un requadre on posar el nom del perfil, així com els botons per guardar-lo i cancel·lar-lo. Per tant, només cal introduir el text desitjat i guardar els canvis.

Nom del perfil:

Il·lustració 82. Opcions per afegir perfil

Un cop guardar, el perfil s'enregistrarà en el sistema i podrà ser consultat en un futur.

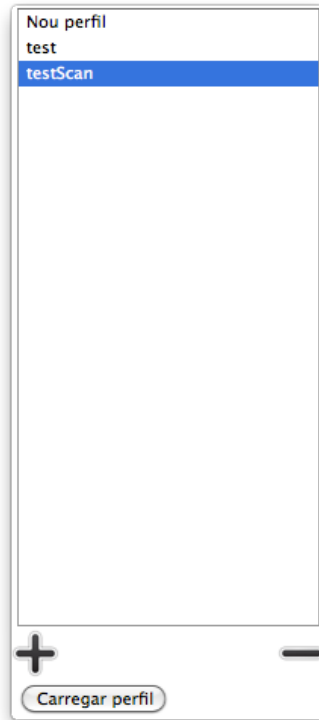


Il·lustració 83. Llistat de perfils amb el nou perfil enregistrat

Carregar perfil

Primerament, cal accedir als perfils, tal i com s'explica en l'entrada "Accedir als perfils".

Un cop es disposa del panell amb la informació relativa als perfils, per carregar un perfil es seleccionarà del llistat aquell perfil guardat prèviament que es desitja carregar. Un cop seleccionat, apareixerà a la part inferior del panell un botó amb el text "Carregar perfil"; al prémer-lo, es tancarà el panell i es carregarà la configuració desitjada.

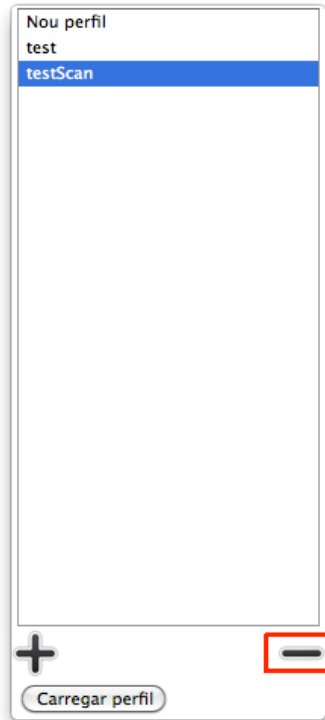


Il·lustració 84. Opció per carregar perfils

Eliminar perfil

Primerament, cal accedir als perfils, tal i com s'explica en l'entrada "Accedir als perfils".

Un cop es disposa del panell amb la informació relativa als perfils, per eliminar un perfil es seleccionarà del llistat aquell perfil guardat prèviament que es desitja carregar. Un cop seleccionat, es prem el botó situat al extrem dret inferior del panell, representat amb un "-". Al prémer-lo, s'eliminarà el perfil seleccionat.

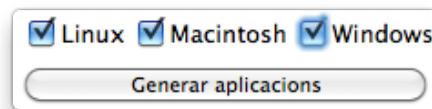


Il·lustració 85. Eliminar perfil del sistema

Generar aplicació usuari

Per generar l'aplicació d'usuari, primerament s'ha de prémer el botó situat a la part inferior dreta del panell d'administració. El botó porta el nom "Generar aplicació usuari".

Al prémer-lo, es desplegarà un panell amb tres caselles associades a cadascun dels Sistemes Operatius que la interfície gràfica suporta. Al activar qualsevol de les tres caselles, s'activarà el botó "Generar aplicacions".



Il·lustració 86. Opcions de generació d'aplicació d'usuari

Per tant, cal primer de tot marcar aquells Sistemes Operatius per als quals es desitja generar l'aplicació de l'usuari final i després prémer el botó "Generar aplicacions". Aquesta acció crearà a l'escriptori de l'usuari una carpeta amb el nom de cadascun dels sistemes operatius desitjats, i dins d'aquesta carpeta els fitxers corresponents a l'aplicació.

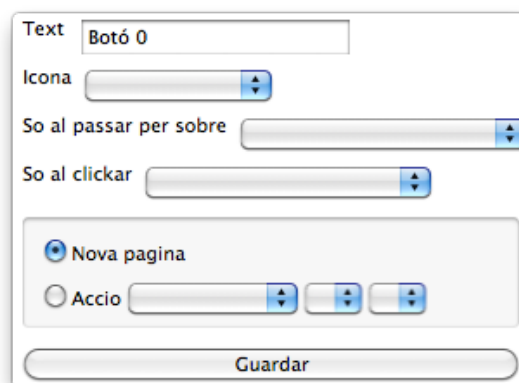


Il·lustració 87. Versions de client generades per la interfície gràfica

Botons

Aquesta secció del document presenta totes aquelles opcions de configuració que afecten a cadascun dels botons adaptables. Aquests botons seran usats pel client final per interaccionar amb els diferents dispositius domòtics.

Per accedir a les opcions dels botons, només cal situar-se sobre qualsevols dels botons situats a l'àrea de previsualització de la interfície gràfica i prémer-lo. Això obrirà un panell a l'alçada del botó seleccionat amb les diferents opcions a realitzar, tal i com es veuran a continuació:



Il·lustració 88. Opcions de configuració d'un botó

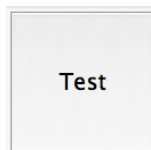
Modificar text boto

Per modificar el text d'un botó, cal situar-se sobre el quadre de text situat a la part superior del panell, sota el títol "Text" i modificar-lo pel text que es consideri adient.



Il·lustració 89. Modificar text botó

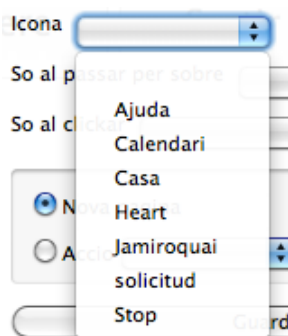
Un cop finalitzat, sols resta prémer el botó guardar per confirmar i comprovar els canvis realitzats en la interfície.



Il·lustració 90. Text del botó modificat

Afegir imatge

Per modificar la imatge d'un botó, cal situar-se sobre el desplegable situat a la part superior del panell, sota el títol "Icona". Al prémer-lo es desplega un llistat amb totes les imatges disponibles en el sistema; només cal seleccionar la imatge que es desitgi. Al seleccionar una imatge, s'ofereix la possibilitat d'amagar el text



Il·lustració 91. Insertar imatge al botó

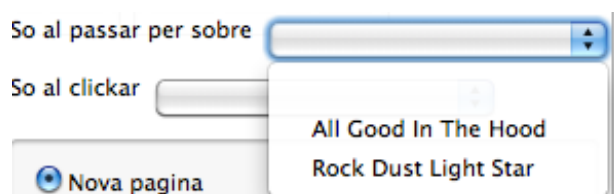
Un cop finalitzat, sols resta prémer el botó guardar per confirmar i comprovar els canvis realitzats en la interfície.



Il·lustració 92. Botó després d'afegir una imatge

Definir so al seleccionar el botó

Per modificar el comportament d'un botó i fer que reproduïxi un so al passar el punter per sobre d'ell, cal situar-se sobre el desplegable amb el títol "So al passar per sobre". Al prémer-lo es desplega un llistat amb tots els sons disponibles en el sistema; només cal seleccionar el so que es desitgi.

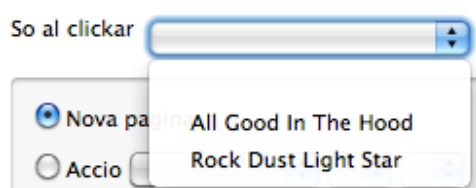


Il·lustració 93. Selecció de so al passar per sobre del botó

Un cop finalitzat, sols resta prémer el botó guardar per confirmar els canvis realitzats en la interfície. Cal recordar que els sons no es reproduiran mentre es treballi amb la versió d'administració; si es que es reproduiran, però, si es fa una vista prèvia de la interfície o quan es generi la versió per a l'usuari final.

Definir so al clicar el botó

Per modificar el comportament d'un botó i fer que reproduïxi un so al prémer-lo, cal situar-se sobre el desplegable amb el títol "So al prémer". Al prémer-lo es desplega un llistat amb tots els sons disponibles en el sistema; només cal seleccionar el so que es desitgi.



Il·lustració 94. Opcions de so al clicar el botó

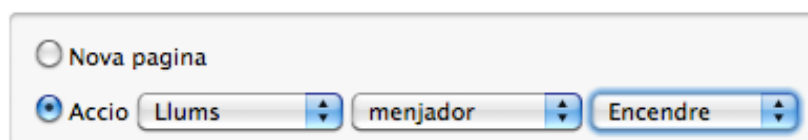
Un cop finalitzat, sols resta prémer el botó guardar per confirmar els canvis realitzats en la interfície. Cal recordar que els sons no es reproduiran mentre es treballi amb la versió d'administració; si es que es reproduiran, però, si es fa una vista prèvia de la interfície o quan es generi la versió per a l'usuari final.

Definir acció del botó

Per modificar el comportament d'un botó i fer que executi una acció determinada al prémer-lo, cal situar-se a la part inferior del panell, sobre el requadre previ al botó "Guardar".

En aquest requadre hi ha dos opcions a escollir:

- Nova pàgina: Si es selecciona aquesta opció, al prémer el botó es navegarà a una pantalla nova. També implica que si s'escull aquesta opció es crearà una nova entrada al llistat de pantalles (zona esquerra de la interfície gràfica) amb el nom del botó.
- Acció: Aquesta opció indica al sistema que al prémer el botó es desitja interactuar amb un dels dispositius domòtics del sistema. Per fer-ho, es disposa de tres desplegable. El primer desplegable indica el tipus de dispositiu que es vol associar (llums, televisor, etc.), un cop es selecciona el dispositiu es carregen els altres dos llistat. El segon llistat serveix per indicar la localització del dispositiu, ja que un tipus de dispositiu pot estar en varies localitzacions alhora (per exemple, les llums es troben al menjador, rebedor, cuina, habitacions...). I finalment, el tercer desplegable permet indicar al sistema que es desitja fer amb el dispositiu seleccionat (en el cas de les llums, obrir-les o apagar-les).



Il·lustració 95. Opcions d'acció d'un botó

Un cop finalitzat, sols resta prémer el botó guardar per confirmar els canvis realitzats en la interfície. Cal recordar que no es realitzaran les accions del botó mentre es treballi amb la versió d'administració; si es que s'executaran, però, si es fa una vista prèvia de la interfície o quan es generi la versió per a l'usuari final.

Navegació

Aquesta secció del document presenta aquelles opcions que permeten navegar entre les diferents visualitzacions que ofereix el sistema; ja sigui per configurar cadascuna de les pantalles, com per comprovar el resultat de la configuració actual.

Per accedir a les opcions de navegació, cal situar-se sobre la zona de navegació, situada a l'extrem esquerra de la interfície gràfica.

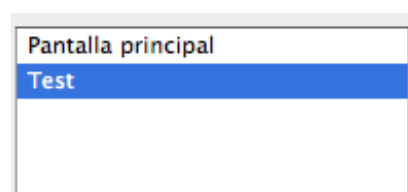


Il·lustració 96. Zona de Navegació

Seleccionar pantalla

Per seleccionar una pantalla i poder-la visualitzar i configurar, cal situar-se sobre el llistat que conté cadascuna de les pantalles existents en la configuració actual del sistema.

Un cop localitzat el llistat, només cal seleccionar l'entrada corresponent a la pantalla que es vol carregar per a que es carregui el seu contingut en la zona de previsualització.

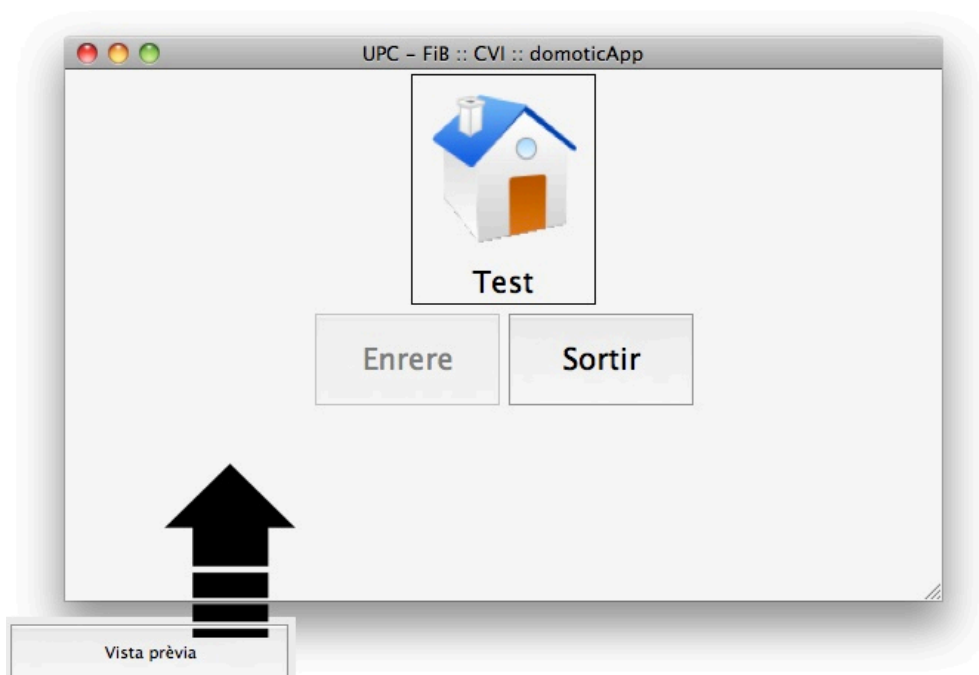


Il·lustració 97. Selecció pantalla

Vista prèvia del resultat

Aquesta funcionalitat permet a l'usuari administrador obtenir una perspectiva del conjunt de l'aplicació tal i com està configurada en el moment de prémer el botó, comprovant així si el comportament del sistema s'ajusta a l'esperat o si cal modificar algunes de les configuracions definides.

Per accedir a aquesta funcionalitat, cal situar-se sobre el botó que es troba a l'extrem inferior de la zona de previsualització i prémer-lo. Això obrirà una nova finestra que només mostrarà a pantalla completa la zona de previsualització amb tots els elements comportant-se segons el perfil actual en el sistema.



Il·lustració 98. Resultat de la "Vista Prèvia"

Interfície de client (usuari final)

Pantalla

L'usuari de tipus Client té com objectiu usar la interfície gràfica amb la finalitat de manipular els diferents dispositius domòtics que hi ha instal·lats a la casa.

A continuació es mostren les diferents funcionalitats que ofereix la interfície per aquest tipus d'usuaris:

Accionar botó

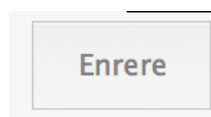
Aquesta funcionalitat permet a l'usuari interactuar amb els diferents dispositius de forma directa. La interfície mostra un seguit de botons , i al prémer un que està configurat per a gestionar un dispositiu domòtic, aquest envia la informació al dispositiu per a que executi l'acció associada al botó (per exemple, obrir les llums del menjador).

Cal comentar que hi ha dos tipus d'accions que pot fer un botó al accionar-se:

- Per un cantó es pot comunicar amb el dispositiu domòtic associat per realitzar una tasca (tal i com s'ha comentat fins ara).
- També es pot configurar un botó per a que carregui una nova pantalla, que contindrà un seguit de botons propis, així com una distribució dels mateixos i un colors de ressaltat també propis, tot i que no necessàriament diferents als de la pantalla anterior.

Navegar per les pantalles

Aquesta funcionalitat és molt similar a l'anterior, "Accionar botó". Totes dues s'activen al prémer un botó; tot i que en aquest cas es tracta d'un botó especial, situat sempre a la part esquerra de l'extrem inferior de la interfície gràfica.



Il·lustració 99. Tornar a la pantalla origen

La funcionalitat d'aquest botó es la de carregar la pantalla origen, si es que la pantalla actual difereix de la pantalla amb la que es carrega la interfície.

Sortir de l'aplicació

Aquesta funcionalitat és molt similar a l'anterior, "Accionar botó". Totes dues s'activen al prémer un botó; tot i que en aquest cas es tracta d'un botó especial, situat sempre a la part dreta de l'extrem inferior de la interfície gràfica.



Il·lustració 100. Sortir de l'aplicació

La funcionalitat d'aquest botó es la de tancar la interfície gràfica i sortir de forma efectiva de l'aplicació.

14.4. Índex d'il·lustracions

Il·lustració 1. Diagrama d'actors.....	34
Il·lustració 2. Classificació dels casos d'ús.....	35
Il·lustració 3. Casos d'ús 'Gestionar aplicació'	36
Il·lustració 4. Casos d'ús 'Gestionar perfils'	40
Il·lustració 5. Casos d'ús 'Gestionar recursos'	43
Il·lustració 6. Casos d'ús 'Obtenir dispositius'	45
Il·lustració 7. Casos d'ús "Gestionar aplicació (client)"	46
Il·lustració 8. Cas d'ús configurar pantalles.....	48
Il·lustració 9. Cas d'ús consultar pantalla	49
Il·lustració 10. cas d'ús configurar botons.....	49
Il·lustració 11. Cas d'ús definir escaneig	49
Il·lustració 12. Cas d'ús vista prèvia del resultat	50
Il·lustració 13. Cas d'ús Generar aplicació d'usuari.....	50
Il·lustració 14. Cas d'ús Carregar perfil	51
Il·lustració 15. Cas d'ús Afegir perfil	51
Il·lustració 16. Cas d'ús Eliminar perfil.....	52
Il·lustració 17. Cas d'ús Afegir recurs	52
Il·lustració 18. Cas d'ús Eliminar recurs.....	53
Il·lustració 19. Cas d'ús Obtenir dispositius	53
Il·lustració 20. Cas d'ús Carregar perfil	54
Il·lustració 21. Cas d'ús Visualitzar pantalla.....	54
Il·lustració 22. Cas d'ús Activar dispositiu	55
Il·lustració 23. Model conceptual referent a la vista de la interfície gràfica.....	57
Il·lustració 24. Model conceptual referent al controlador de la interfície gràfica.....	58
Il·lustració 25. Model conceptual referent al domini de la interfície gràfica.....	59
Il·lustració 26. Diagrama distribució física del sistema.....	63
Il·lustració 27. Distribució espai en interfície gràfica.....	66
Il·lustració 28. Configuració de pantalla.....	67
Il·lustració 29. Configuració escaneig.....	67
Il·lustració 30. Administració aplicació	67
Il·lustració 31. Navegació entre pantalles	68
Il·lustració 32. Previsualització pantalla. Configuració per defecte	69
Il·lustració 33. Configuració botó	70
Il·lustració 34. Directori de fitxers.....	72
Il·lustració 35. Fitxer de preferències.....	73
Il·lustració 36. Definició de l'esquelet de l'aplicació	74
Il·lustració 37. Càrrega inicial del sistema.....	74
Il·lustració 38. Lectura de dades de disc	75
Il·lustració 39. Crida a codi Java (DLA) des de Javascript.....	76
Il·lustració 40. Connexió amb el DLA.....	76
Il·lustració 41. Codi enquesta fitxer XML al DLA	77

Il·lustració 42. Javascript processant dades retornades via Java	77
Il·lustració 43. Pantalla inicial d'interfície d'administrador	79
Il·lustració 44. Codi d'invocació Javascript a un panell XUL.....	80
Il·lustració 45. Definició d'un panell en XUL	80
Il·lustració 46. Desplegament en interfície del panell invocat per Javascript.....	81
Il·lustració 47. Captura de petició per guardar les dades realitzat per la interfície al controlador	82
Il·lustració 48. Modificació de l'estat d'un botó	83
Il·lustració 49. Resultat modificació botó interfície	83
Il·lustració 50. Diagrama Gantt.....	90
Il·lustració 51. Repartiment recursos humans/cost.....	93
Il·lustració 52. Interfície hardware: ratolí.....	105
Il·lustració 53. Interfície hardware: joystick tradicional	106
Il·lustració 57. Interfície hardware: pantalla tàctil.....	109
Il·lustració 58. Interfície software: punter	110
Il·lustració 59. Interfície hardware: escaneig	111
Il·lustració 60. Opcions de pantalla	113
Il·lustració 61. Introducció manual # botons.....	114
Il·lustració 62. Funcions semi automàtics per al # botons.....	114
Il·lustració 63. Barra d'ajustament de la mida.....	114
Il·lustració 64. Barra d'ajustament de l'espaiat entre botons.....	114
Il·lustració 65. Modificar color de fons de l'aplicació	115
Il·lustració 66. Activar alertes visuals	115
Il·lustració 67. Ressaltat de botons mitjançant colors	116
Il·lustració 68. Opcions d'escaneig.....	116
Il·lustració 69. Opcions d'escaneig en funcionament	117
Il·lustració 70. Activar opcions d'escaneig.....	117
Il·lustració 71. Opcions d'escaneig.....	118
Il·lustració 72. Opcions d'administració.....	118
Il·lustració 73. Panell de gestió de sons.....	119
Il·lustració 74. Opcions per afegir sons.....	119
Il·lustració 75. Eliminar sons.....	120
Il·lustració 76. Panell imatges disponibles en el sistema.....	121
Il·lustració 77. Opcions per afegir imatges.....	121
Il·lustració 78. Eliminar imatge del sistema	122
Il·lustració 79. Accés a la gestió de perfils.....	122
Il·lustració 80. Panell amb els perfils disponibles en el sistema.....	123
Il·lustració 81. Botó per afegir perfils.....	123
Il·lustració 82. Opcions per afegir perfil.....	124
Il·lustració 83. Llistat de perfils amb el nou perfil enregistrat.....	124
Il·lustració 84. Opció per carregar perfils.....	125
Il·lustració 85. Eliminar perfil del sistema	126
Il·lustració 86. Opcions de generació d'aplicació d'usuari	126

Il·lustració 87. Versions de client generades per la interfície gràfica.....	127
Il·lustració 88. Opcions de configuració d'un botó	127
Il·lustració 89. Modificar text botó	127
Il·lustració 90. Text del botó modificat.....	128
Il·lustració 91. Insertar imatge al botó.....	128
Il·lustració 92. Botó després d'afegir una imatge.....	128
Il·lustració 93. Selecció de so al passar per sobre del botó.....	129
Il·lustració 94. Opcions de so al clicar el botó.....	129
Il·lustració 95. Opcions d'acció d'un botó.....	130
Il·lustració 96. Zona de Navegació	131
Il·lustració 97. Selecció pantalla.....	131
Il·lustració 98. Resultat de la "Vista Prèvia"	132
Il·lustració 99. Tornar a la pantalla origen.....	133
Il·lustració 100. Sortir de l'aplicació	134

14.5. Índex de taules

Taula 1. Cas d'ús "Configurar pantalles"	36
Taula 2. Cas d'ús "Consultar pantalla"	37
Taula 3. Cas d'ús "Configurar botons"	37
Taula 4. "Definir escaneig"	38
Taula 5. Cas d'ús "Vista prèvia del resultat"	39
Taula 6. Cas d'ús "Generar aplicació d'usuari"	39
Taula 7. Cas d'ús "Carregar perfil"	40
Taula 8. Cas d'ús "Afegir perfil"	41
Taula 9. Cas d'ús "Eliminar perfil"	42
Taula 10. Cas d'ús "Afegir recurs"	43
Taula 11. Cas d'ús "Eliminar recurs"	44
Taula 12. Cas d'ús "Obtenir dispositius"	45
Taula 13. Cas d'ús "Iniciar Sistema"	46
Taula 14. Cas d'ús "Visualitzar pantalla"	47
Taula 15. Cas d'ús "Activar dispositius"	47
Taula 16. Costs en recursos humans	92
Taula 17. Cost software desglossat.....	94
Taula 18. Costs Hardware i Software.....	94
Taula 19. Cost total del projecte.....	94