



# **Aplicació gamificada per estalvi energètic en el sector hoteler**

**TREBALL FINAL DE GRAU**  
**GRAU EN ENGINYERIA INFORMÀTICA**  
**TECNOLOGIES DE LA INFORMACIÓ**

**AUTOR:**  
**Marc Capdevila Canadell**

**DIRECTOR I PONENT:**  
**Pau Fonseca i Casas**  
Professor UPC-FIB

## **Resum**

L'objectiu d'aquest T.F.G., és la creació d'una aplicació mòbil per a l'estalvi energètic en hotels. Per tal de portar a cap aquest estalvi energètic, cal adaptar el consum de recursos de les habitacions al model optim de consum calculat per Necada. Aquesta guia per portar al client cap al model optim d'hàbits de consum, es farà a través de suggeriments mostrats en l'aplicació.

Això es pretén realitzar dissenyant i desenvolupant l'aplicació software i preparant la connectivitat amb Necada de cara a un futur.

## **Abstract**

The purpose of this T.F.G., is the creation of a mobile application for energy saving in hotels. In order to carry out this savings, the consumption of resources in the rooms must be adapted to an optimum consumption model calculated by Necada. This guide aims to bring customers to this optimal model of consumption habits, and will be pursued through suggestions shown furtherly in the application.

It is intended to develop this idea by designing the software application and preparing the connectivity with Necada for the upcoming future.

## Glossari

**API:** Es l'abreviatura de "Application Programming Interface". Una API es un conjunt de comandes, funcions, protocols i objectes que els programadors poden utilitzar per a crear software o interaccionar amb un sistema extern. Normalment, una API proveu als desenvolupadors amb comandes estàndard per a realitzar tasques repetitives i/o comunes, per a que no tinguin que escriure el codi des de zero.

**Necada:** Es una infraestructura híbrida que dona suport a l'execució d'un model de simulació al nuvol per tal d'ajudar a optimitzar els recursos que s'utilitzin en qualsevol etapa del cicle de vida d'un edifici.

**APP:** Aplicació mòbil.

**Spin-off:** Les spin-off són iniciatives empresarials promogudes per membres de la comunitat universitària, que es caracteritzen per basar-se sumi activitat en l'explotació de nous processos, productes o serveis a partir del coneixement adquirides i els resultats obtinguts en la mateixa Universitat.

# Índex de contingut

1. Introducció	1
1.1 Introducció	1
1.2 Formulació del problema	1
1.3 Context	3
1.3.1 Estat del art	3
1.3.2 Objectius	5
1.3.3 Parts interessades	5
2. Abast del projecte	6
2.1 Definició de l'abast	6
2.2 Reptes	8
3. Metodologia de treball	8
3.1 Recursos	9
3.2 Validació dels objectius	11
3. Planificació Temporal	12
3.1 Tasques i dependències	12
4. Valoració d'alternatives i pla d'acció	14
5. Integració de coneixements	15
6. Identificació de lleis i regulacions	16
7. Pressupost	16
7.1. Identificació dels costos	16
7.2. Estimació dels costos	18
7.3. Control de gestió	19
8. Sostenibilitat i compromís social	19
8.1. Dimensió econòmica	19
8.2. Dimensió social	19
8.3. Dimensió ambiental	20
9. Aplicació	20
9.1. Requisits Aplicació	20
9.2. Disseny	21

9.3. Implementació	26
10. Back-end.	28
10.1 Requisites back-end	28
10.2 Disseny back-end	30
10.3. Implementació back-end	35
11. CONCLUSIONS	36
12. VALORACIÓ PERSONAL	36
Bibliografia	37
Annex – Diagrama de Gantt	39

## Índex de figures

1. Imatge de l'Àrtic al 2018.	2
2. Imatge del creixement de l'economia espanyola de "El País".	3
3. Logotip de Necada.	4
3. Logotip de GRAPHISOFT.	4
4. Arquitectura dels sistemes de comunicació.	7
5. Organització del sprint en Asana.	10
6. Pantalla d'inici, dispositiu Android.	21
7. Pantalla d'inici, dispositiu IOs.	21
8. Pantalla principal del Administrador, dispositiu IOs.	22
9. Pantalla Rooms List, dispositiu IOs.	23
10. Pantalla Admin settings, dispositiu IOs.	23
11. Pantalla Detail Room Page, dispositiu IOs.	24
12. Pantalla Periods List, dispositiu IOs.	24
13. Pantalla principal d'un client, dispositiu IOs.	25
14. Pantalla de detall d'un consell, dispositiu IOs.	25
15. Esquema de l'arquitectura típica d'una aplicació Xamarin Forms.	26
16. Tipus de Pàgines de Xamarin Forms.	27
17. Tipus de contenidors de Xamarin Forms.	27
18. Arquitectura final del back-end.	29

## Índex de taules

Taula 1: Recursos hardware.	11
Taula 2: Recursos software.	11
Taula 3: Tasques del projecte.	13
Taula 4: Costos hardware segons la seva amortització en el projecte.	18
Taula 5: Hores i costos dels diferents rols per a les diferents tasques del projecte.	18
Taula 6: Pressupost final.	19

# 1. Introducció

## 1.1 Introducció

Aquest projecte es un Treball de Fi de Grau de la Facultat d'Informàtica de Barcelona (Universitat Politècnica de Catalunya), concretament de l'especialitat de tecnologies de la informació.

El projecte és ofert pel professor Pau Fonseca Casas, a partir d'un projecte més gran que es Necada<sup>[1]</sup> al qual se li vol ampliar les seves funcionalitats de cara al sector hoteler.

L'objectiu del projecte és oferir una solució tecnològica en forma d'aplicació mòbil gamificada que permeti als usuaris d'un hotel consumir eficientment els recursos d'aquest a canvi d'una recompensa econòmica en la factura.

Abans de seguir anem a definir que es Necada?

“Necada és una infraestructura híbrida que admet l'execució d'un model de simulació en entorns de núvol, granja o escriptori. L'objectiu d'aquesta infraestructura és trobar valors òptims per a diversos paràmetres de construcció i els impactes associats que redueixen la demanda o consum energètic de l'edifici. La infraestructura combina els dos llenguatges formals i la co-simulació per obtenir una definició inequívoca completa del model, assegurant que la implementació final usará les millors eines per simular qualsevol component del model.”<sup>[1]</sup>

o de manera més simplificada:

Necada és una infraestructura híbrida que dóna suport a l'execució d'un model de simulació al nuvol, per tal d'ajudar a optimitzar els recursos que s'utilitzin en qualsevol etapa del cicle de vida d'un edifici.

## 1.2 Formulació del problema

Un dels temes més controvertits en l'actualitat és el Canvi Climàtic i els efectes que té sobre el planeta. És un tema de debat, on els seus efectes ja són notables i amenacen de ser generalitzats i catastròfics. Aquests efectes tenen incidència en els mitjans de subsistència de totes les parts del planeta, en les economies, en la salut i en la seguretat alimentària global.

Alguns d'aquests efectes són: augment del nivell de la mar, onades de calor, grans tempestes, grans sequeres, espècies en extinció, destrucció d'ecosistemes, etc.

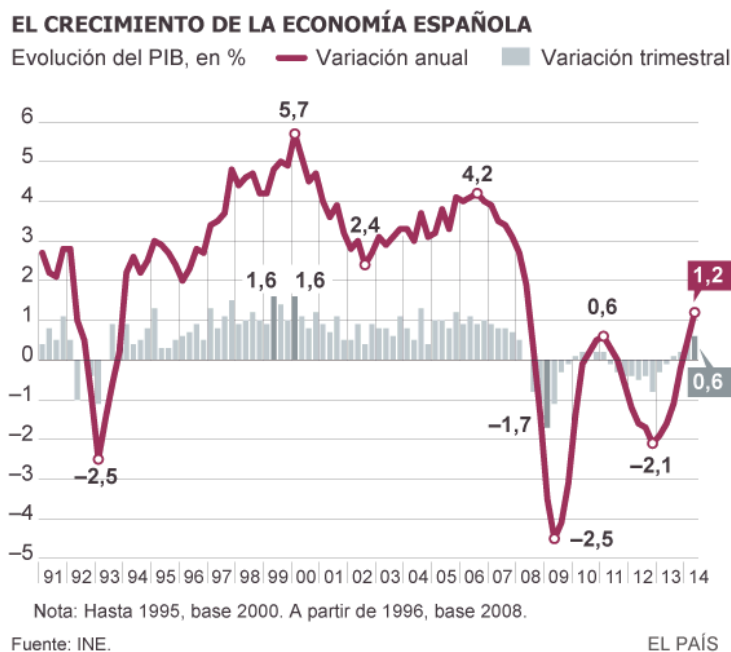


1. Imatge de l'Àrtic al 2018.

Dit això podem dividir la població en tres grans grups: La que intenta lluitar per a combatre aquests efectes, la que lluitaria contra aquests efectes si sabes com i a la que no li interessa el tema i no pretén lluitar contra aquest efectes. Ens interessen els dos primers grups. Si aquests dos grups sabessin quines accions poden realitzar per a consumir de forma eficient els recursos ajudaríem a no malgastar-ne una gran quantitat.

Per altra banda, tenim un dels altres temes més controvertits en l'actualitat, la crisi econòmica.

La crisi econòmica que pateix Europa des de 2008 és la pitjor des de la Gran Depressió dels anys 30. No ha acabat el 2018 i els seus efectes encara s'estan fent notar, tant en les seves conseqüències



per a la població, com en els canvis institucionals i empresarials.



## 2. Imatge del creixement de l'economia espanyola de "El País".

Dins de les conseqüències per a la població, una de les més greus es la pujada dels impostos que junt amb el bloqueig de les pujades de salari ha provocat que moltes famílies haguin "d'apretar-se el cinturó", tinguin que ajustar les despeses.

Per a totes aquestes famílies qualsevol iniciativa que els permeti estalviar-se una part petita o gran de diners serà benvinguda per ells.

Donats aquests dos grans problemes sorgeixen projectes com Necada, que intenten aportar el seu granet d'arena en aquesta lluita.

Necada és un projecte en constant desenvolupament que vol arribar a proporcionar les eines necessàries per a maximitzar l'eficiència que li donem a tots recursos que utilitzem en el cicle de vida dels edificis.

El problema es que Necada actualment fora de l'àmbit acadèmic sol te un "spin-off" que exporta les dades dels models calculats. La resta dels projectes que exporten dades de Necada fins al moment, són projectes de baix nivell, propis i d'àmbit acadèmic per a l'observació dels models.

Així que podem dir que la informació de Necada es veu limitada a un petit grup de persones que tenen accés i que tenen coneixements d'estadística o de llenguatges com R per treure alguna conclusió.

Per això el desenvolupament d'una aplicació mòbil que permeti l'explotació de les dades dels models calculats permetrà transmetre aquesta informació d'una forma comprensible a un rang de public més general.

A més els aspectes de gamificació de l'aplicació junt amb les recompenses en forma de descomptes promouran d'una forma entretinguda i dinàmica que els usuaris de hotels consumeixin els recursos que aquest els hi proporciona d'una forma eficient.

## **1.3 Context**

### **1.3.1 Estat del art**

Abans de desenvolupar un sistema d'aquestes característiques cal revisar en quin punt es troba la solució que volem desenvolupar, si podem aprofitar altra tecnologia ja existent o adaptar-la per a nosaltres.

En el nostre cas com partim d'un projecte més gran, Necada, del qual ja aprofitem la seva tecnologia.



3. Logotip de Necada.

Però és una bona idea partir de Necada? O hi han millors solucions de les quals partir?

Existeix un projecte anomenat Acad<sup>[2]</sup> de Adobe similar però abasta molt menys. Acad a diferencia de Necada no introdueix al model les variables ambientals, socials i econòmiques així com tampoc té cura de les directives ambientals i les normes internacionals en el procés de disseny.

Projectes com GRAPHISOFT EcoDesigner STAR<sup>[3]</sup> però són específics per a un recurs, en aquest cas GRAPHISOFT es específic per a l'optimització de recursos energètics i tampoc arriba a contemplar totes les variables com Necada.

Building Information Model



Building Energy Model

3. Logotip de GRAPHISOFT.

O bé en referència a Necada hi ha un projecte de Final de Carrera<sup>[4]</sup> de la UB, sobre el negoci intel·ligent enfocat a Necada però no tracta el tema en la línia que ho proposem nosaltres.

Per tant podem dir que no hi ha cap projecte d'aquesta índole que creï models virtuals de tot el cicle de vida d'un edifici tenint en compte variables ambientals, socials i econòmiques així com tenint cura de les directives ambientals i les normes internacionals en el procés de disseny.

Per altra banda, hi ha molts edificis que ja utilitzen dispositius IoT (Internet of things), que els hi permeten millorar l'eficiència dels recursos de l'edifici, com per exemple sensors de llum i temperatura que regulen la llum i la temperatura depenent de les dades que estan captant.

Aquests dispositius no són una competència per a Necada sinó que de cara a un futur podrien ser el següent pas en el projecte que avui proposem. Permeten que des de l'aplicació es puguessin activar o desactivar certes automatitzacions depenent de l'usuari.

Però no tot es bo, Necada no maneja dades reals de cap hotel ni ha implementat la recepció de dades de sensors de hotels en la seva infraestructura. Aquesta part del projecte serà teòrica a la espera que s'implementi algun dia, ja que la intenció de fer-ho hi es i esta impulsada per el seu responsable, en Pau Fonseca Casas.

Per tant assumeixo que Necada es la millor solució de la qual partir.

### 1.3.2 Objectius

Un cop vist el problema i avaluat el estat de l'art, podem resumir els objectius no tècnics que el projecte pretén resoldre:

1. Crear consciència de què realment estem malbaratant recursos cars per al planeta i per la nostra butxaca.
2. Informar dels recursos que estem malbaratant i quines accions podem emprendre per a solucionar-ho de forma fàcil i entenedora.
3. Recordar activament a les persones que ho desitgin de què poden millorar l'eficiència dels recursos utilitzats a través de notificacions.
4. Estalviar diners de forma directa o indirecta als clients d'un hotel i als gerents d'aquest.

Aprofundirem més aquests objectius així com la tecnologia que emprarem en el apartat d'Abast del projecte.

### 1.3.3 Parts interessades

A continuació es defineixen les parts interessades en el projecte (a qui va dirigit el producte, qui ho usará i qui es beneficiará dels seus resultats), en el cas del nostre projecte tindrem :

1. **Cadenes hoteleres:** La preparació de l'hotel i la campanya que li donin perquè els usuaris a través d'una sèrie de descomptes utilitzin la nostra aplicació, li pot portar una gran reducció en el consum de recursos que utilitza, que dit d'una altra manera, li pot aportar un gran estalvi econòmic.

2. **Usuaris d'aquestes cadenes hoteleres:** Tot depenent de l'hotel, aquests usuaris podrien rebre descomptes en la seva factura depenent de la utilització que li donin als recursos de l'hotel tot seguint els consells que se'ls hi ha proporcionat a través de la nostra App.
3. **Necada:** Es beneficiarà de dues maneres, per una banda les dades recol·lectades/calculades podran ser servides un públic més general e indirectament es donarà a conèixer el projecte que està portant a cap.
4. **Població:** Indirectament qualsevol persona es beneficia d'aquest o qualsevol projecte que redueixi el consum de recursos del planeta.

## 2. Abast del projecte

### 2.1 Definició de l'abast

L'abast d'aquest projecte com ja hem dit se centra dins del marc del projecte Necada en la millora de l'eficiència dels recursos utilitzats en la vida útil d'un edifici, l'interval entre que l'edifici s'ha construït i l'edifici s'enderrocarà o es remodelarà. En concret en centrem en el sector hotelier però podria ser escalable a qualsevol edifici del qual Necada tingues dades per a poder extreure'n el seu model de simulació.

El principal sistema a construir serà l'aplicació mòbil amb la que interactuaran els usuaris (usuaris de l'hotel i administradors de l'hotel). Per tant haurà de comptar amb les següents característiques:

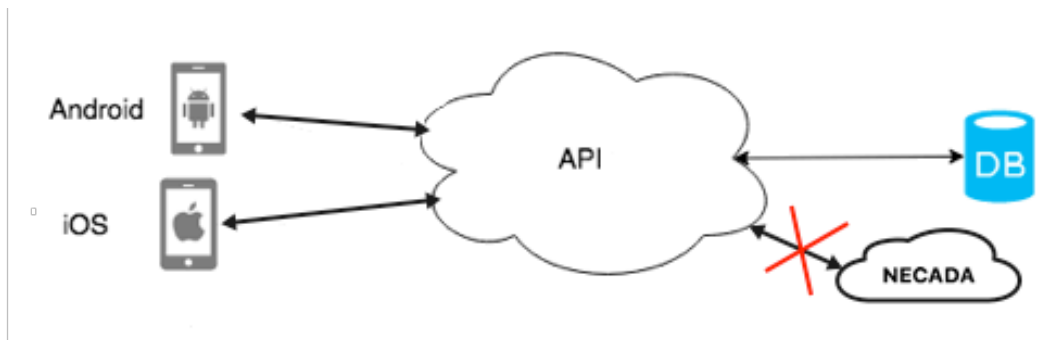
Per als usuaris del hotel (clients), tindran accés a:

- Dades descriptives de l'habitació: hotel, pis, porta, etc.
- Eficiència actual dels recursos.
- Eficiència total dels recursos.
- Consells per a millorar la eficiència.

Per als administradors del hotel (gerents), tindran accés a:

- Dades descriptives de l'hotel:
- Descomptes que s'aplicaran segons l'us de les recomanacions per part dels clients.
- Llistat amb cercador de habitacions.
- Informació de les habilitacions: hotel, pis, porta, usuari, contrasenya, històric de eficiències dels usuaris, etc.

Per donar suport i obtenir les dades necessàries per convertir l'aplicació mòbil en un sistema distribuït ens caldrà implementar un sistema de dades i un servei web adient. Aquest servei web constarà d'una API web (Interfície de Programació d'Aplicacions).



4. Arquitectura dels sistemes de comunicació.

A més intentarem construir la nostra API amb els següents beneficis:

- Construïrem un sistema escalable.
- Construïrem un sistema mantenible.
- Construïrem un sistema fiable.
- Construïrem un sistema segur.

Com podem veure en la figura anterior, no hi ha els dispositius mòbils amb sistema operatiu Windows, ja que he decidit que no val la pena invertir recursos del projecte en implementar l'aplicació en Windows si un dels seus responsables, Joe Belfiore el 8 d'octubre del 2017 va admetre via Twitter que ja no implementarien noves funcionalitats per al sistema operatiu mòbil de Windows.

També podem observar com la connexió amb Necada esta taxada amb una creu en vermella. Com Necada no esta preparat per a les consultes, ni disposa de dades reals d'edificis, ni de sensors que captin dades a temps real, es deixara preparada la connexió per part de la nostra Api de cara a un futur.

Però per part de Necada com ja hem comentat abans caldrà:

- Afegir les variables extres necessàries al model.
- Connectar els sensors del hotel amb el servidor de Necada.
- Implementar les crides d'actualització de dades de Necada cap a la nostra API
- Implementar el endpoints necessaris per a que la nostra API faci consultes a Necada.

Finalment, el projecte es donarà per finalitzat quan s'assoleixin les funcionalitats ofertes pels sistemes mencionats anteriorment.

## 2.2 Reptes

El desenvolupament d'aquest projecte comporta una sèrie de reptes, tant d'escalabilitat, de fiabilitat, de disseny, etc.

En primer lloc, hem de construir un sistema escalable. L'aplicació podria ser usada per qualsevol persona que se l'instal·li al seu dispositiu. Per tant, hem de preveure que el nombre d'usuaris podria créixer fins a nombres prou grans.

En segon lloc, haurem de dissenyar una aplicació usable i visualment atractiva, per a reduir el *bouncerate* dels usuaris, és a dir, el percentatge d'usuaris que es baixen l'aplicació però la desinstalen en un període curt de temps.

Per últim, podem trobar-nos un problema determinat per la quantitat de dades de les quals disposa Necada per a realitzar el seu model de simulació, així com la utilització i l'extracció d'informació de matrius complexes de càlcul, ja que no és una característica en la qual m'hagi format.

## 3. Metodologia de treball

Per dur a terme aquest projecte utilitzarem la metodologia àgil<sup>[8]</sup>, aquesta metodologia es basa en iteracions que en aquest cas seran d'una durada de 2 setmanes en la que es desenvolupen un seguit de tasques.

Cada iteració inclou un seguit de fases que són les següents:

1. Planificació de la feina que s'ha de dur a terme segons les histories d'usuari definides (Backlog).
2. Anàlisi dels requisits (segons els costos temporals i la disponibilitat de temps de l'equip de desenvolupament).
3. Disseny
4. Implementació de les tasques
5. Retroalimentació i millores.
6. Documentació.

L'objectiu de cada iteració és el d'implementar parts de l'aplicació que siguin funcionals, és a dir, que no tinguin errors.

Dins de les metodologies àgils hi ha diferents mètodes que poden ser aplicats com poden ser: Scrum, Extreme Programming, Kanban, etc.

En aquest cas utilitzarem la metodologia Scrum<sup>[9]</sup>, aquest mètode es basa en adoptar una estratègia de desenvolupament incremental. Basa la qualitat del producte en equips autogestionats i es caracteritza per la superposició en les diferents fases en comptes d'una implementació seqüencial.

L'elecció d'aquest mètode es basa en una experiència prèvia en el treball fora de la FIB i de les comparacions dels resultats obtinguts en referència a metodologia en cascada utilitzada en la FIB en algun projecte.

Per tant eh decidit adoptar els aspectes principals del mètode Scrum, ja que la flexibilitat pel que fa a canvis en el projecte durant les diferents iteracions i un constant flux de "feedback" entre el client (en aquest cas serà el mateix director del projecte) i el desenvolupador crea un millor resultat redueix els riscos als quals pot estar exposat el projecte.

Primer de tot el que farem serà dividir el projecte en iteracions segons les setmanes que tingui disponibles per dur a terme la realització del projecte.

Tot seguit dividirem el projecte en casos d'ús i els hi assignarem una prioritat segons els requisits del client, en aquest cas com ja hem dit el nostre client serà el director del projecte.

Cal remarcar que a mesura que s'implementin iteracions del projecte segurament s'aniran reavaluant la prioritat de les tasques.

Un cop acabada cada iteració es farà una presentació al client per tal de detectar possibles errors, incorporar millores i modificar les tasques o les prioritats.

### 3.1 Recursos

Per a la realització del projecte s'empraran diferents recursos hardware i software, així com recursos humans, que en aquest cas sol hi participará de forma activa dues persones:

- El director del projecte, que assumirà els rols de director i de client.
- El autor (jo mateix), que assumiré tots els altres rols dins de la metodologia Scrum.

L'eina per a l'organització temporal és Asana<sup>[10]</sup>, que a mes permet compartir el calendari amb altra gent (en aquest cas trobo que és bona idea que el director tingui accés i sàpiga en quin punt es troba el projecte).



#### 5. Organització del *sprint* en Asana.

A més a més, utilitzarem 2 eines més que són GitHub<sup>[11]</sup> per al control de versions del codi i finalment Skype<sup>[12]</sup> i Gmail<sup>[13]</sup> per a una constant comunicació amb el director.

Pel que fa a les tecnologies que utilitzarem per dur a terme totes les funcionalitats i implementar tots els sistemes mencionats, ens cenyirem als requeriments tecnològics requerits pel professor Pau Fonseca Casas, ponent del projecte, i condició necessària del desenvolupament del projecte: l'arquitectura del servidor la implementarem amb .NET Core 2.0, la última versió del framework de .NET de Microsoft. I per donar suport a una integració continua utilitzarem els serveis d'Azure i Visual Studio com a IDE.

En el cas de l'aplicació després de rebuçar assoleixo que la millor opció és Xamarin<sup>[5]</sup>, ja que el projecte havia de ser en .net i volem que sigui multiplataforma, hi ha altres opcions com react natiu<sup>[6]</sup> i ionic<sup>[7]</sup> però són en JavaScript.

Xamarin genera tres projectes, un per Android, un per iOS i un per Windows. En el nostre cas sol generarem el de Android e iOS. A més, genera un projecte comú que compartirà gran part del codi. Aquest projecte pot ser desenvolupat en varis tipus, però l'he generat en .netstandard, així utilitzarem les últimes tecnologies de .NET Core.

Per els elements front-end, Xamarin compta amb una llibreria anomenada Xamarin Forms que ens permet desenvolupar en el projecte comú de forma comuna per a diferents plataformes fen servir els seus elements.

A més Xamarin Forms una vegada compilat transforma les aplicacions a codi Natiu. La única pega es que no és tant eficient com si s'hagues fet amb natiu, li carga una mica més de pes al projecte, un 20% més.

Pel que fa a la base de dades, he decidit utilitzar una base de dades NoSQL ja que te una escalabilitat horitzontal, és capaç de créixer en número de màquines en lloc de tenir que residir en grans màquines, fet que ens permet complir el objectiu d'escalabilitat.

Per tant, tenin en compte que ja usem Microsoft azure per a la API, per temes de compatibilitat, he elegit escollir Azure cosmos bd de Microsoft azure, que usa Mongo DB com a base de dades.

En les següents taules podrem observar els recursos hardware i software generals que s'han utilitzat i amb quina finalitat.



RECURS	FINALITAT
MacBook Pro (Retina, 15 pulgadas, High Sierra) amb processador 2,2 GHz Intel Core i7	Ordinador personal el qual s'usarà per al desenvolupament del projecte.
Asus GL553VD-DM065T con i7, 8GB amb processador Intel® Core™ i7-7700HQ	Ordinador personal de la feina necessari per a la creació i test de l'aplicació en Windows.
Apple iPhone 7	Mòbil personal en el qual es testejarà l'aplicació.
Nexus 6	Mòbil d'un familiar en el qual es testejarà l'aplicació.

Taula 1: Recursos hardware.

RECURS	FINALITAT
Skype	Software per a les reunions online.
GitHub	Software per al control de versions del codi.
SourceTree <sup>[14]</sup>	Software que facilita el control de versions i la seva disposició.
Asana	Software per a l'organització del projecte i les seves tasques.
Gantter <sup>[15]</sup>	Software per a la creació dels diagrames de Gantt.
Visual Studio <sup>[16]</sup>	Entorn software per al desenvolupament del front-end i back-end.
Gmail	Software per a les comunicació amb el director.
Servei App Service de Microsoft Azure.	Servidor i Servei de Microsoft Azure en el qual podem pujar la API.
Azure cosmos bd.	Servidor de base de dades de Microsoft Azure en el qual podem pujar la nostra base de dades.

Taula 2: Recursos software.

### 3.2 Validació dels objectius

Com es tracta d'un projecte que utilitzarà la metodologia Scrum tindrà els seus mètodes de validació i correcció d'errors, com és la revisió dels contingut en finalitzar la iteració com ja hem dit abans. Això ens permet que el client, en el nostre cas el director del projecte, a cada iteració podrà corregir, descartar o ampliar depenent del "feedback" que ell tingui.

Cal remarcar que no sempre s'ha de fer cas del que diu el client al peu de la lletra, ja que molts no saben exactament el que volen. Així que hem d'actuar una mica com de consultors escoltant clarament les seves exigències i proposant solucions que realment siguin vàlides per a ell.

Finalment per al tema de càlculs i extracció de dades de les matrius proporcionades pel model creat per Necada és crearan tests a totes les funcions que s'utilitzin.

### 3. Planificació Temporal

La planificació temporal del projecte estava delimitada entre els mesos de febrer i juny, ambdós inclosos però per qüestions personals i professionals aliens al projecte s'ha tingut de reduir el temps diari de 8h inicials que havia proposat a 3h diàries.

Donat aquest canvi tant gran s'ha recorregut a demanar una extensió del plaç d'entrega, que deixa una planificació temporal entre els mesos de febrer i octubre.

El projecte s'iniciarà el 12 de febrer amb la presentació de (GEP – Gestió de Projectes) i és té previst finalitzar-lo el 08 d'octubre deixant així uns dies de marge per la preparació de la presentació del projecte per el que pugui passar.

Tot i això les hores comptabilitzades en el projecte s'han pres a “grosso modo” de manera que ja és té en compte en cada tasca una possible desviació o marge que és podrà usar per al desenvolupament en cas que el termini no sigui el correcte o en cas contrari, per a la millora.

#### 3.1 Tasques i dependències

El projecte s'ha dividit en 5 tasques generals que al seu temps aquestes tindran subtasques. En la taula següent podem veure la seva distribució i duració. També ho podem observar de forma temporal i amb les dependències en el Gantt inclòs en l'Annex 1.

TASCA	HORES
<b>1. Disseny i planificació de la plataforma i les seves funcionalitats</b>	<b>140</b>
1.1 Definir el projecte (GEP – Gestió de Projectes)	80
1.2 Establir les funcionalitats de l'aplicació	15
1.3 Dissenyar gràficament el prototip de l'aplicació	40
1.4 Proposta de l'aplicació	5
<b>2. Implementació del prototip de l'aplicació</b>	<b>113</b>
2.1 Aprenentatge del llenguatge .net i XAML (Xamarin)	60
2.2 Implementació del prototip pactat	30

TASCA	HORES
2.3 Presentació del prototip	3
2.4 Estudiar la retroalimentació rebuda	5
2.5 Aplicar millores	15
<b>3. Implementació del back-end de la plataforma</b>	<b>255</b>
3.1 Aprenentatge de la tecnologia usada en el back-end	25
3.2 Estudi de l'estat actual del back-end (NECADA)	25
3.3 Aprenentatge del tractament de dades que és farà servir	25
3.4 Estudi i aprenentatge de connexions segures.	30
3.5 Implementació del back-end.	120
3.6 Realització de tests i millores	30
<b>4. Unió i validació de la connexió front/back-end</b>	<b>40</b>
4.1 Unió front/back-end	30
4.2 Realització de tests i millores	10
<b>5. Documentació</b>	<b>100</b>
5.1 Redactar la documentació final	70
5.2 Preparar la presentació	30
<b>TOTAL</b>	<b>648</b>

Taula 3: Tasques del projecte.

En el Gantt adjunt en l'Annex 1 es poden visualitzar les diferents tasques en el calendari. Doncs bé, anem a profunditzar i a explicar el perquè d'aquestes dependències.

En la primera tasca general com és sobreentén, no tenim cap dependència, però dins de les seves subtasques sí. En concret deixa'n de banda la tasca de GEP la resta tenen una dependència en cascada que passa per establir les funcionalitats, dissenyar gràficament el prototip i realitzar la proposta.

La segona tasca general depèn de la primera tasca general, ja que per a implementar el prototip primer tindriem que tindre la proposta (subtasca 1.4) i les seves subtasques tenen una dependència en cascada, ja que es tracta de l'estudi de les eines a utilitzar, desenvolupament, retroalimentació i millora.

La tercera tasca, desenvolupament del back-end general depèn de la subtasca 1.2, ja que primer haurem de tenir la definició de les funcionalitats de l'aplicació. En les seves subtasques podem observar que la implementació té com a dependències les subtasques 3.1, 3.2, 3.3 i 3.4 que són tasques d'aprenentatge de tecnologies a usar i com és lògic aquesta tasca és la dependència de la subtasca de test i millora.

La quarta tasca, unió i validació de la connexió front/back-end depèn com és lògic de les tasques de desenvolupament de front i back-end. Pel que fa a les seves subtasques podem observar que tenen una dependència en cascada, ja que és implementació de la unió i test del qual s'hagi implementat.

Finalment la tasca de documentació depèn de la penúltima tasca, ja que es realitzarà al final de tot.

## 4. Valoració d'alternatives i pla d'acció

A continuació es valoraran els diferents riscos que podrien arribar a aparèixer durant el transcurs del projecte en les diferents tasques en les quals l'hem dividit.

En la primera tasca, Disseny i planificació de la plataforma i les seves funcionalitats, trobem únicament un risc bastant unit a la resta de tasques, que es tracta de superar les hores establertes en la planificació, tot i que ja s'ha tingut en compte un cert número d'hores de marge en cada tasca.

En la segona tasca el risc que trobem és el que el "feedback" del disseny sigui molt negatiu, cosa que implicaria la superació del temps previst.

En la tercera tasca, com ja vam comentar el risc que hi ha i considero que és el més gran. És el de la comprensió de l'estat actual de Necada i de l'extracció de les dades per al front-end pel simple fet que aquesta rama de tractament de dades no és una branca en la qual m'hagi especialitzat.

En la quarta tasca no es preveu cap risc, ja que s'han assignat les suficients hores per als seus possibles inconvenients i sol es tractaria de desenvolupar els últims detalls de la unió front/back-end.

I finalment, en l'última tasca, la de documentació tampoc es valoren riscos, ja que durant el curs de GEP ja s'hauran redactat documents i creades presentacions dels quals ja tindrem un "feedback" de les coses a millorar i vigilar.

Un cop identificats els diferents riscos de les tasques cal traçar un pla d'acció i establir diferents alternatives per tal de minimitzar les desviacions temporals degudes a aquests riscos.

Pel que fa al risc de planificació l'únic que podem fer és donar cert número d'hores de marge en cada tasca i un marge final d'uns dies com ja s'ha tingut en compte.

Pel que fa al risc del "feedback" del disseny sigui molt negatiu es valorarà demanar assessorament a un expert en disseny o UX depenent del que factor que sigui molt negatiu.

I finalment per al risc de la comprensió de l'estat actual de Necada i de l'extracció de les dades per al front-end es valorarà també demanar assessorament a un expert en estadística.

## 5. Integració de coneixements

Els coneixements necessaris per a poder desenvolupar aquest TFG de manera satisfactòria s'han adquirit de dues fonts. La primera es la universitat i les assignatures que he cursat durant el Grau, i la segona es l'experiència professional adquirida en els últims anys de carrera.

Les assignatures que més coneixements m'han aportat per a que pugui desenvolupar aquest projecte correctament han sigut les següents:

- BD: Capacitat de creació i gestió de base de dades.
- AD: Capacitat d'avaluació de les diferents formes de comunicació i elecció de la mes indicada.
- ADEI: Tractament de dades en R, útil per a la creació de una idea mental de les dades servides per necada.
- ER: Capacitat de determinar quin sistema s'ha de construir.
- PTI: Capacitat de plantejar una proposta de projecte i refinar-la.
- PI: Capacitat de decidir si la instal·lació de la infraestructura TIC proposada de forma teòrica seria factible.
- SI: Capacitat d'identificació i prevenció de possibles vulnerabilitats del sistema.
- TCI: Capacitat de comprensió de codificacions.

Gràcies a l'experiència laboral de diferents empreses, els coneixements adquirits han sigut els següents:

- Saber identificar quins son els elements clau per a garantir un nivell alt de seguretat en les comunicacions entre aplicacions.
- Aprendre metodologies de desenvolupament segur.
- Capacitats avançades d'aprenentatge de nous llenguatges.
- Procediments per a una correcta integració de tecnologies.

## 6. Identificació de lleis i regulacions

En el nostre projecte hem identificat dues lleis que ens afectarien:

**GDPR**, per les seves sigles en anglès (General Data Protection Regulation), o RGPD per les seves sigles en Català (Reglament General de Protecció de Dades) és la nova normativa que regula la protecció de les dades dels ciutadans que viuen en la Unió Europea. La qual de forma general ve a dir, que hauràs de donar el teu consentiment inequívoc perquè les empreses puguin fer servir les dades si ets ciutadà europeu. A més, t'hauran de dir quines dades estan utilitzant, com les estan tractant, per què i qui és la persona responsable de les mateixes.

**Llei de Serveis de la Societat de la Informació i Comerç Electrònic (LSSI-CE)**, per garantir el compliment amb la Directiva sobre la privacitat electrònica que exigeix:

- L'acceptació dels termes i condicions d'ús per part de l'usuari, després de haver-se-li facilitat informació clara i completa sobre la identitat del prestador de serveis, així com informació relativa de la cessió a tercers, i de les dades vinculades a la utilització de l'aplicació instal·lada.
- El consentiment previ per part de l'usuari per a la instal·lació de l'APP i per a la utilització d'altres aplicacions de què disposa el propi dispositiu (per exemple: la llista de contactes, correu electrònic, etc.) i de les que la nova aplicació requereix per al seu propi ús.

## 7. Pressupost

### 7.1. Identificació dels costos

En la realització de les tasques mencionades anteriorment tenim les despeses següents:

1. Despeses en recursos humans.
2. Despeses en hardware.
3. Despeses en software.
4. Despeses indirectes.

Cal recalcar que la identificació i càlcul d'aquestes despeses està basat en el diagrama de Grantt de l'annex 1.

En el cas de les despeses en recursos humans tenim que en aquest projecte intervindran un cap de projecte un dissenyador, un programador i un tester tots els papers seran implementats per mi mateix i intervindran en les diferents fases del projecte depenent del seu paper.

En referència a despeses hardware tenim, com ja hem comentat anteriorment, un ordinador MacBook Pro, un ordinador Asus, un Iphone 7 i un Nexus 6.

En quan a despeses en software obtenim que la gran majoria ens surt gratis, ja sigui per estar inclòs en el preu dels recursos hardware, gràcies a ser estudiant el podem utilitzar gratuïtament o bé perquè es tracta de software lliure. L'única excepció que hi ha es en referència als serveis de Azure.

Pel que fa a la API, la plataforma ens ofereix un tipus de servei anomenat App Service, que ens permet publicar petits serveis. Aquests App Services són serveis en els que Azure ja ens permet publicar una aplicació en .Net i són gratuïts. Ja que la nostra aplicació en un principi no tindrà un gran flux d'usuaris i per tant és difícil sobrecarregar el servidor, inclourem el App Service en un anomenat Service Plan, que és la màquina física on s'allotgen els serveis.

El cost de mantenir una màquina d'aquest tipus, en la seva configuració més baixa és zero. Azure ofereix un tipus molt bàsic de Service Plan gratuït, que comparteix recursos amb màquines d'altres usuaris. Per començar i fer proves ens serà suficient, si més endavant necessitem escalar la màquina podem fer-ho fàcilment.

Per altra banda tenim Azure Cosmos bd que ofereix com un servei de base de dades multimodal distribuït a tot el món, Azure Cosmos DB facilita la creació d'aplicacions escalables i de gran capacitat a escala global. Ofereix diversos models de coherència ben definits, garanteix valors de latència de lectura i escriptura en mil·lisegons d'un sol dígit en el percentil 99, així com un 99,999% d'alta disponibilitat amb allotjament múltiple a qualsevol part del món. Però, no es gratuït, encara que sol pagues per el seu us.

Escollint la configuració de 1Gb i escriptura en una sola regió obtenim que el preu estimat al més es de 32,35 €.

Per a desenvolupar i testejar una aplicació iOS/Android no cal més que un ordinador però per a publicar-les cal llicències (comptes de desenvolupadors) . Pe el cas de Apple el compte de desenvolupador té un cost de 99 dòlars a l'any (uns 80 euros) i en el cas de Android el compte de desenvolupador té un cost de 25 dòlars (uns 20 euros), en un únic pagament.

I finalment pel que fa a despeses indirectes obtenim que al pressupost si li hauria de sumar una quantitat aproximada de 700€ (preu mitjà per a una oficina a la ciutat de Barcelona d'uns 120 m) per lloguer d'oficina i uns 1500€ per la possible aplicació dels diferents plans de contingència (assessors, hores extra, serveis d'Azure i llicències de desenvolupament Apple i Android).

## 7.2. Estimació dels costos

En relació als costos hardware que estimem que tindrem obtenim que:

Fase	Rol				Cost
	Cap de projecte (40€/h)	Dissenyador (30€/h)	Programador (25€/h)	Tester (20€/h)	
Disseny i planificació	80h	40h	20h	-	4.900 €
Implementació del prototip	8h	5h	100h	-	2.970 €
Implementació del back-end	10h	-	235h	10h	6.475 €
Unió i validació	-	-	35h	5h	975 €
Documentació	40h	-	60h	-	3.100 €
<b>TOTAL</b>	<b>138h</b>	<b>45h</b>	<b>450h</b>	<b>15h</b>	<b>18.420 €</b>

Taula 4: Costos hardware segons la seva amortització en el projecte.

Les hores comptabilitzades per al càlcul de l'amortització és la d'una jornada laboral de 3 hores al dia, 216 dies a l'any.

La vida útil dels recursos és la mitjana estàndards que tenen aquests, en portàtils és d'uns 3 anys i en telèfons intel·ligents és d'uns dos anys en el camp del desenvolupament.

Pel que fa als costos en recursos humans obtenim que:

Fase	Rol				Cost
	Cap de projecte (40€/h)	Dissenyador (30€/h)	Programador (25€/h)	Tester (20€/h)	
Disseny i planificació	80h	40h	20h	-	4.900 €
Implementació del prototip	8h	5h	100h	-	2.970 €
Implementació del back-end	10h	-	235h	10h	6.475 €
Unió i validació	-	-	35h	5h	975 €
Documentació	40h	-	60h	-	3.100 €
<b>TOTAL</b>	<b>138h</b>	<b>45h</b>	<b>450h</b>	<b>15h</b>	<b>18.420 €</b>

Taula 5: Hores i costos dels diferents rols per a les diferents tasques del projecte.



Finalment obtenim que el cost total seria de:

RRHH	Harware	Software	Indirectes	TOTAL
18.420 €	344,945 €	0 €	700€ + 1.500€	20.964,945 €

Taula 6: Pressupost final.

### 7.3. Control de gestió

El control de gestió de les despeses del projecte es basarà en una demanda de "feedback" continua amb el director del projecte en les quals s'avaluarà si s'estan complint els temps i els recursos aplicats són els que s'havien predit o si s'ha d'aplicar alguna mesura del pla d'acció perquè no s'estan complint els temps o els recursos aplicats en aquell moment.

A més en cada tasca general s'actualitzarà el pressupost i el número d'hores que s'han invertit en ella per tal de veure si podem tenir més marge en les altres tasques en el cas que s'hagin gastat menys recursos i menys temps del predit, o per contra si estem superant la planificació i hem d'aplicar els plans d'acció i hem de prendre més mesures al respecte.

## 8. Sostenibilitat i compromís social

### 8.1. Dimensió econòmica

Pel que fa a dimensió econòmica aquest projecte està especialment dissenyat perquè es redueixin els costos econòmics en els hotels. Quant parlo de reduir és costos econòmics dels hotels hem refereixo tant a la gerència de l'hotel, és a dir, als costs que tindrà l'hotel com als clients d'aquest que estalviaran a mesura de la seva implicació.

En aquest apartat el projecte rep una qualificació de 9 punts, ja que es considera que es realitzarà amb el menor temps possible i sense assumir grans riscos a causa de les contingències esmentades anteriorment així com els grans avantatges econòmics que implica per al hotel i clients.

### 8.2. Dimensió social

La dimensió ambiental és també un dels punts en el qual està basat el projecte, ja que el que intentarà és maximitzar els recursos que s'estan usant en els hotels. Cal dir que aquest maximitzar, dependrà com sempre de la implicació del client.

Per altra banda en els recursos utilitzats en el projecte, el gran cost serà en recursos humans i pel que representa a recursos hardware, aquests no s'han comprat expressament per al projecte, sinó que ja eren de la meua propietat.

I per finalitzar la dimensió ambiental, en el càlcul de cost s'ha tingut una vida útil de 3 anys en els portàtils i de 2 anys en els telèfons intel·ligents, ja que és la que se'ls hi dona en la teoria, però a la pràctica aquests recursos a no ser que tinguin un problema greu se'ls hi allarga la vida útil. Potser ja no en temes de desenvolupament però podria passar a ser un ordinador personal per a qualsevol persona. Per totes aquestes raons el projecte rep una puntuació de 9 en la dimensió social.

### **8.3. Dimensió ambiental**

La dimensió ambiental és també un dels punts en el qual està basat el projecte, ja que el que intentarà és maximitzar els recursos que s'estan usant en els hotels. Cal dir que aquest maximitzar, dependrà com sempre de la implicació del client.

Per altra banda en els recursos utilitzats en el projecte, el gran cost serà en recursos humans i pel que representa a recursos hardware, aquests no s'han comprat expressament per al projecte, sinó que ja eren de la meua propietat.

I per finalitzar la dimensió ambiental, en el càlcul de cost s'ha tingut una vida útil de 3 anys en els portàtils i de 2 anys en els telèfons intel·ligents, ja que és la que se'ls hi dona en la teoria, però a la pràctica aquests recursos a no ser que tinguin un problema greu se'ls hi allarga la vida útil. Potser ja no en temes de desenvolupament però podria passar a ser un ordinador personal per a qualsevol persona.

Per tant en aquest aspecte rep una qualificació de 9 punts, per tots els aspectes comentats anteriorment.

## **9. Aplicació**

### **9.1. Requisits Aplicació**

Per a desenvolupar el front-end he utilitzat el framework Xamarin per útil desenvolupar les aplicacions d'Android e iOS. Xamarin m'ha permès reutilitzar tot el codi de lògica de l'aplicació, i gran part del codi que gestiona la part visual, a excepció d'algunes parts que requerien d'una delegació als projectes natius.

Ho he desenvolupat també utilitzant Visual Studio per Mac i la versió de Xamarin 7.6.5, que és bastant estable.

Xamarin genera tres projectes, un per Android i un per iOS. A més, genera un projecte comú que compartirà gran part del codi. Aquest projecte pot ser desenvolupat en varis tipus, però l'he generat en .netstandard, així utilitzarem les últimes tecnologies de .NET Core i s'ha pogut compartir codi amb el back-end i en un futur amb Necada si es necessita.

Per a la part visual, com hem dit abans, s’ha utilitzat Xamarin Forms, que presenta tot un seguit de components per a crear les vistes en el projecte comú i que posteriorment en la compilació traduirà aquests components a natius en els respectius projectes.

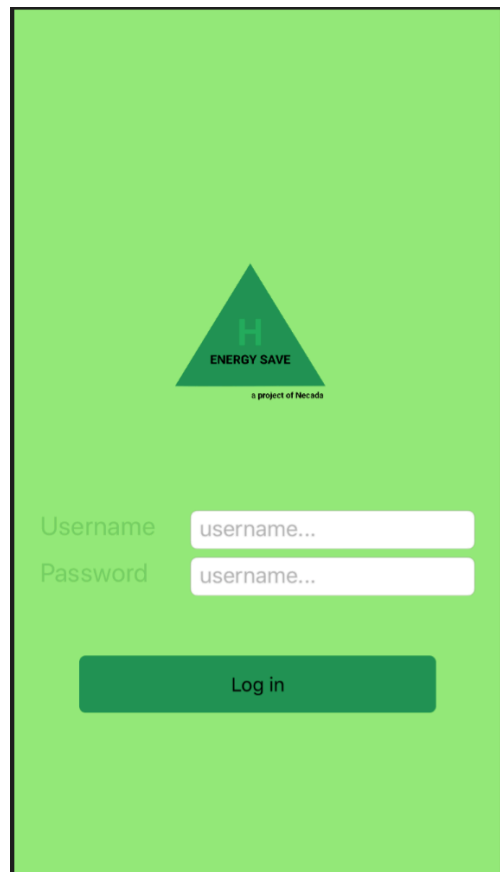
Cal dir que he enfocat molt el projecte a que tingui comunicacions segures, en el cas de la comunicació entre l’aplicació i la Api es fa mitjançant https i controlo el accés a recursos mitjançant tokens(JWT).

## 9.2. Disseny

L’aplicació consta de varis dissenys, depenent de la plataforma de destí. Aquest és un exemple de la pàgina d’inici en Android i IOs:



6. Pantalla d’inici, dispositiu Android.



7. Pantalla d’inici, dispositiu IOs.

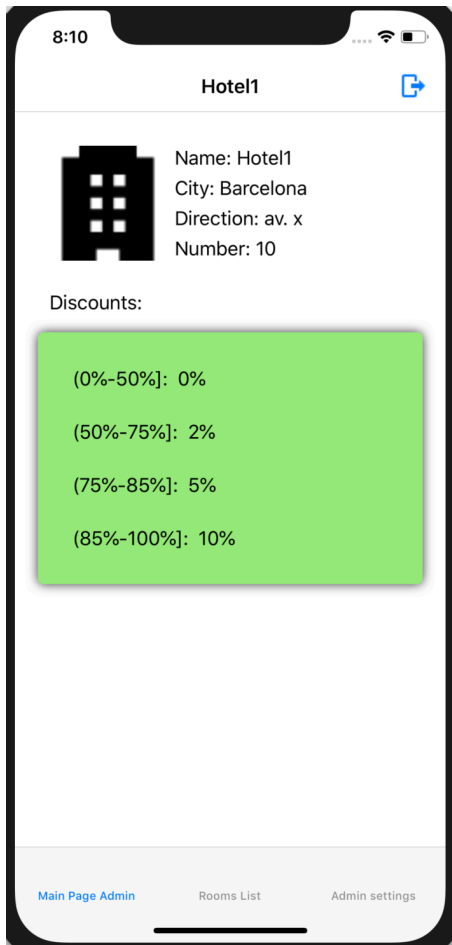
Com podem observar cal autenticar-se per a entrar i no hi ha enllaç per a registrar-se. Això es deu a que els usuaris seran creats per administrador de l’aplicació un cop es configuri l’hotel amb sensors per a enviar dades a Necada i entri a usar el projecte.

Un hotel tindrà un compte com a administrador i un compte per a cada habitació connectada a Necada.

Als usuaris de l’hotel se’ls hi servira el usuari i contrasenya de la seva habitació, que el administrador de l’hotel podrà consultar en el seu compte.

Un cop entrin a l’aplicació depenent si som administradors o usuaris veuran diferents dissenys. Per a simplificar, mostrarem les captures de l’aplicació en IOs.

En el cas de l’Administrador, un cop entri podrà observar:



Pantalla inicial que conté les dades del hotel i els descomptes oferts. Els descomptes s’indiquen en percentatge i estan assignats a un rang d’eficiència dels recursos.

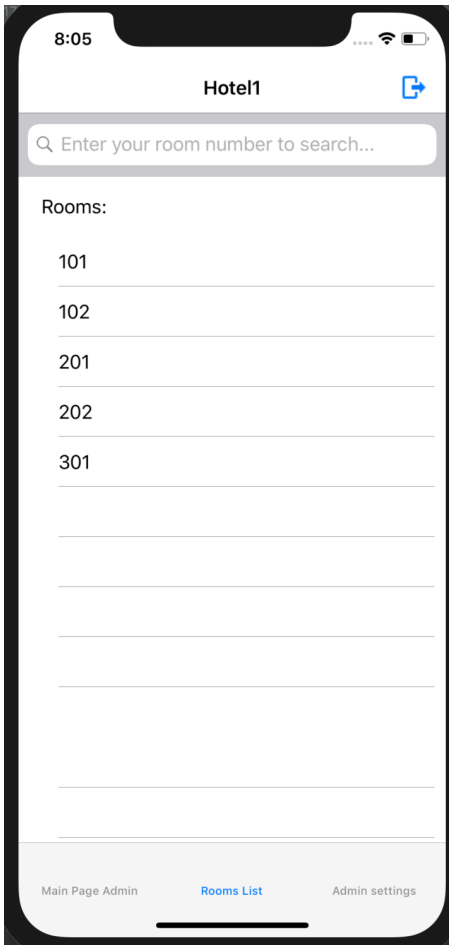
Un cop el usuari del hotel hagi acabat l’estada el hotel depenent del rang d’eficiència que hagi fet dels recursos se li aplicara el descompte corresponent.

En la part de baix de la imatge podem observar un menú per el qual podrà navegar.

I finalment en la part superior al costat del nom te un boto per a sortir del perfil.

8. Pantalla principal del Administrador, dispositiu IOs.

Si canviem de pestanya en el menú a la de “Rooms list” podrem observar la llista d’habitacions de les que disposa el hotel.



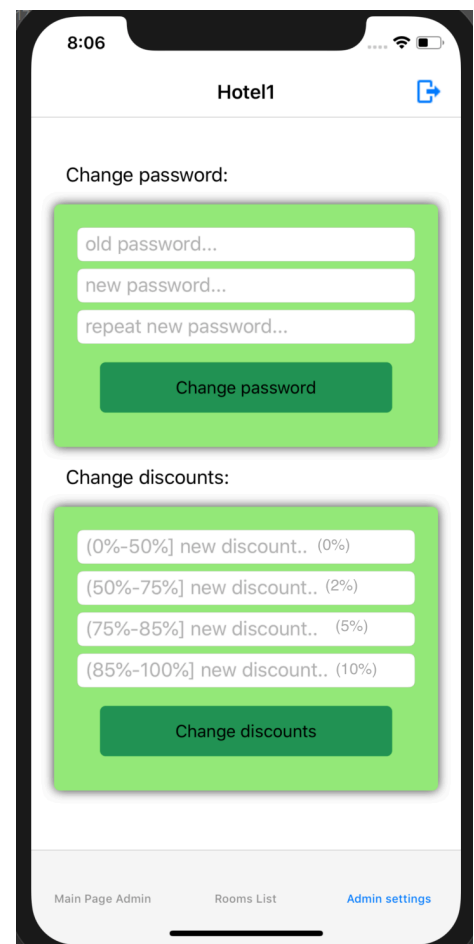
En la part superior podrem veure un cercador que filtra per el nom de les habitacions i si en premem una navegarem a la pagina d'informació de l'habitació.

9. Pantalla Rooms List, dispositiu IOs.

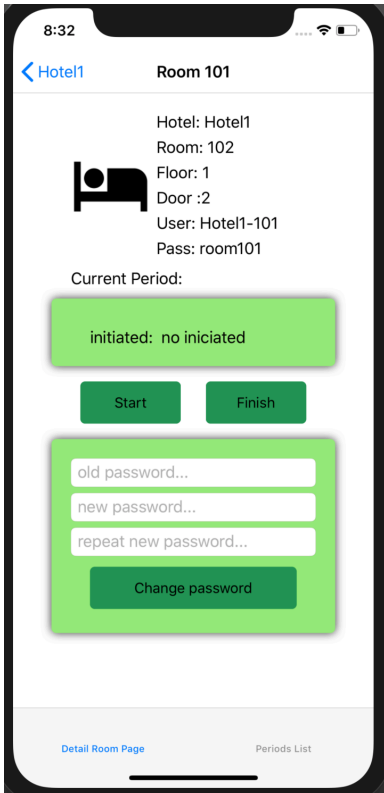
Per altra banda, si accedim a la pestanya de “Admin settings”, observarem dos formularis. El primer per a canviar la contrasenya del compte d'administrador i el segon, a sota, per a modificar els descomptes oferts per l'hotel.

En el del canvi de contrasenya, si un dels camps esta buit, la contrasenya antiga no es correcta o les dos noves contrasenyes no son iguals, no s'efectuara el canvi.

En el formulari dels descomptes sol tenim que modificar el que vulguem canviar. S'indica el valor actual entre parèntesis en el *placeholder*.



10. Pantalla Admin settings, dispositiu IOs.



Si tornem enrere a la llista de habitacions i en seleccionem una, accedim com hem dit a la pagina de detall de l’habitació. On podem observar les dades d’aquesta, el moment que es va iniciar un període, si en hi ha algun actiu i dos botons per iniciar o parar els períodes.

S’iniciara un període quan el client obtingui l’habitació i es parara quan la deixi.

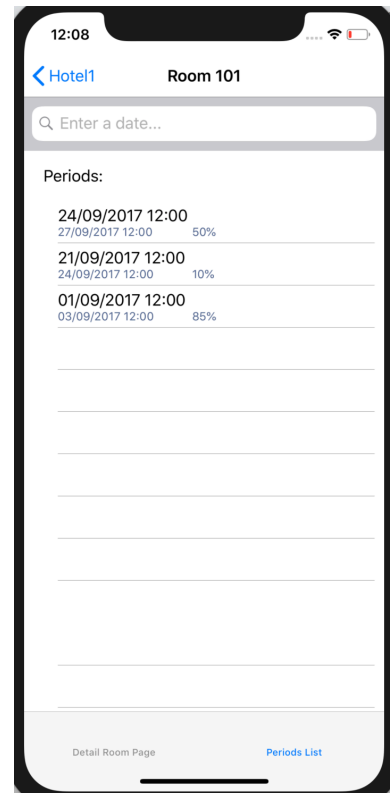
Finalment en la part baixa del contingut de la pantalla observarem un formulari de canvi de contrasenya. Aquest formulari es per a canviar la contrasenya de l’usuari de l’habitació.

11. Pantalla Detail Room Page, dispositiu IOs.

Si canviem en el menú de la pantalla a “Periods List” observarem tots els períodes registrats en l’habitació amb cercador a la part més alta.

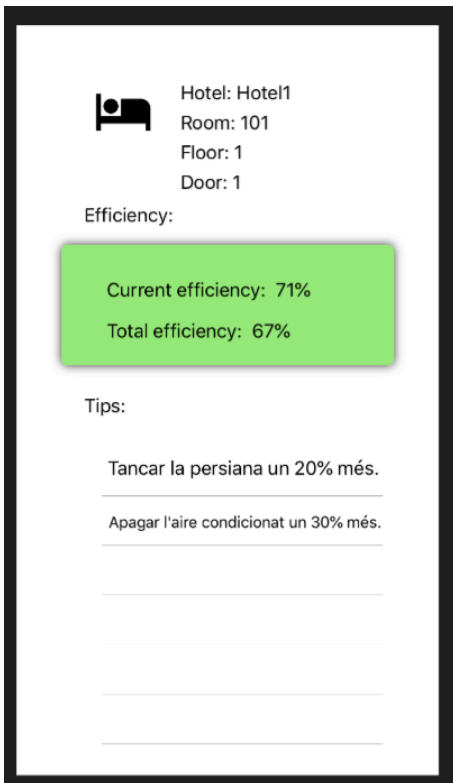
Els períodes estan formats per una data d’inici la de d’alt i a baix la data de finalització junt amb l’eficiència total dels recursos durant el període.

El cercador busca per la data d’inici en forma de text.



12. Pantalla Periods List, dispositiu IOs.

Per el contrari si entrem com a client del hotel, amb el usuari i contrasenya que ens facilitarà l’hotel, podrem observar la següent pantalla:

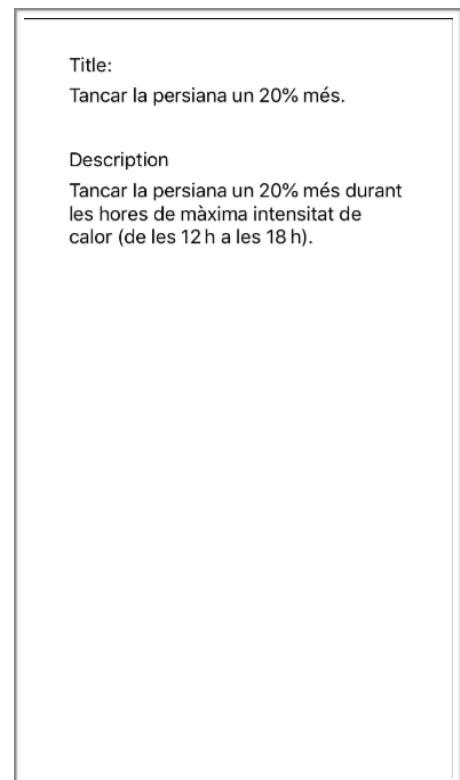


En aquesta pantalla es visualitzen les dades de l’habitació, l’eficiència actual, l’eficiència total ( des de que tenim l’habitació) i un llistat amb els consells per a millorar l’eficiència que els hi estem donant als recursos de l’habitació.

Si premem un del consells de la llista, anirem a para la vista en detall del consell.

13. Pantalla principal d’un client, dispositiu IOs.

En aquesta pantalla de vista en detall del consell, s’ens explicara de forma extensa i clara que es recomana fer per a millorar l’eficiència que els hi estem donant als recursos de l’habitació.



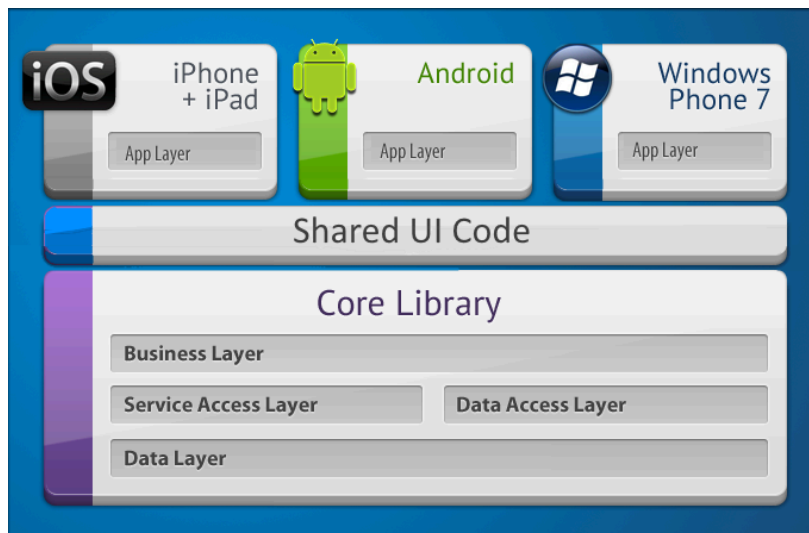
14. Pantalla de detall d’un consell, dispositiu IOs.

### 9.3. Implementació

L'arquitectura usada en el desenvolupament del front-end, ha sigut una arquitectura en 4 capes el model MVC (Model-View-Controller) . El que mostra la figura 15 és que tenim un projecte o llibreria compartida que conté els components de les capes de negoci, serveis i dades.

D'altra banda, tenim diferents aplicacions per a cada tipus de client, que utilitzen les llibreries i components nadius de la User Interface de cada plataforma. Les aplicacions client accedeixen al codi compartit per interactuar amb les capes de negoci, serveis i dades.

Però entremig, Xamarin forms ha introduït una capa per a compartir elements en la creació de les vistes.

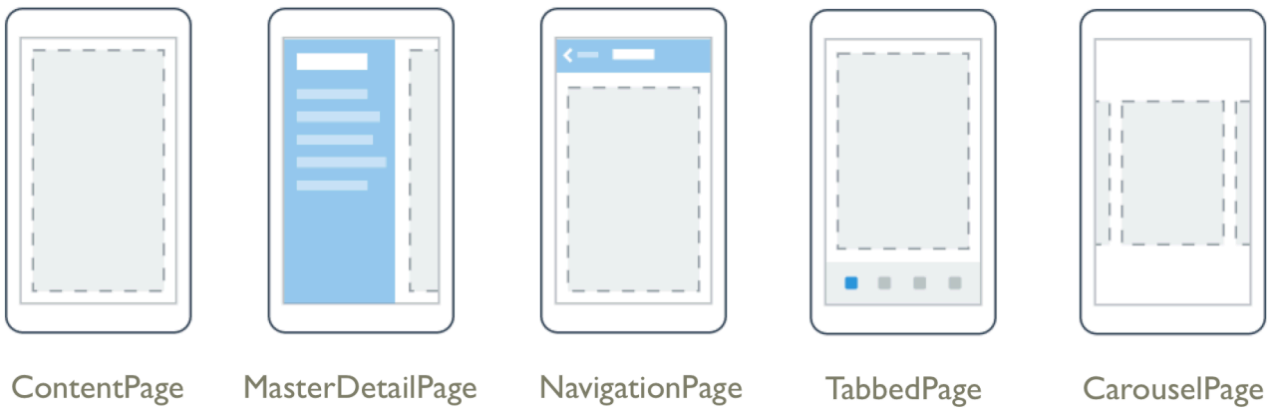


15. Esquema de l'arquitectura típica d'una aplicació Xamarin Forms.

La capa “Shares UI Code” on implementarem les vistes amb els seus propis elements. Alguns dels seus elements més importants són i/o que hem usat són:

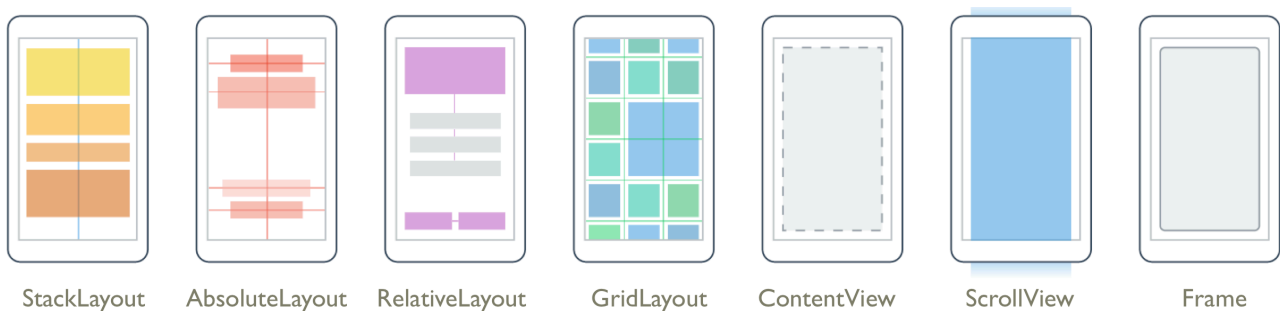
**Les pàgines**, són controls contenidors que representen un Activity en Android i un View Controller en iOS.





16. Tipus de Pàgines de Xamarin Forms.

**Els layouts**, són contenidors d'altres layouts o vistes. Els hem utilitzat per posicionar els controls dins de les pàgines.



17. Tipus de contenidors de Xamarin Forms.

Views o controls que hem utilitzat:

- **BoxView** : Control per pintar un rectangle farcit de color.
- **Button** : Control tipus botó.
- **Entry** : Control per editar text en una sola línia.
- **Image** : Control per mostrar imatges.
- **Label** : Control per mostrar text només de lectura.
- **ListView** : Control per mostrar una llista.
- **Picker** : Control per seleccionar un element d'una llista.
- **ActivityIndicator** : Control visual per avisar que alguna cosa està en curs sense informar el progrés.
- **SearchBar** : Control de cerca.
- **TableView** : Estructura contenedora de columnes i cel·les.

**Cel·les predefinides** per facilitar l'ús en llistes i TableViews.

- **EntryCell** : Cel·la especialitzada amb un label per al títol i un entry per a la introducció de dades.
- **SwitchCell** : Cel·la especialitzada amb un label per al títol i un switch per seleccionar.
- **TextCell** : Cel·la especialitzada amb dos labels, un amb el títol i l'altre amb la descripció.
- **ImageCell** : Cel·la especialitzada amb una imatge i dues labels, un per al títol i un altre per a la descripció.

Pel que fa a la seguretat, com ja hem dit abans les comunicacions amb la API usen el protocol JWT, (JSON Web Token). És un estàndard per transferir tokens segurs amb els permisos incrustats. D'aquesta forma, en autenticar a l'usuari, només s'ha de desfer el token per saber quin usuari està fent la petició i quins permisos té.

Aquest token conté 3 parts separades per un punt, concatenades en la forma “capçalera.cos.signatura”:

- **Capçalera:** A la capçalera s'hi indica el tipus de token i quin algoritme s'ha utilitzat per codificar-lo. La capçalera es codifica en Base64.
- **Cos:** Conté la informació útil. En el nostre cas usem: *username*, *hotel* i *room*. El cos es codifica en Base64.
- **Signatura:** La signatura és un hash generat a partir de la capçalera codificada, el cos codificat, una clau secreta i l'algoritme introduït a la capçalera.

Amb aquest token permetem que l'API identifiqui l'usuari que està fent la petició, i li doni accés o el refusi segons si el usuari esta autoritzat a usar el recurs o no.

## 10. Back-end.

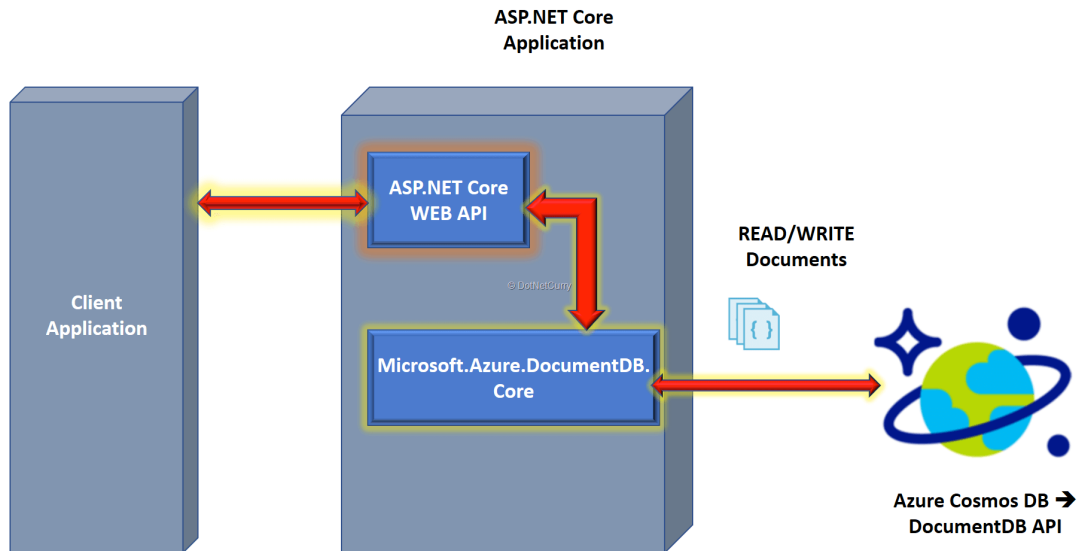
### 10.1 Requisits back-end

Per realitzar el projecte s'ha desenvolupat servei que es una API amb la tecnologia .NET Core 2.0, de Microsoft, que ens permet desenvolupar aplicacions web tant des de Windows, com MacOS com Linux. En el meu cas, he desenvolupat amb el sistema MacOS.

He utilitzat l'entorn de desenvolupament Visual Studio per Mac, que és gratuït. És la nova versió del Visual Studio de Windows, però pel sistema MacOS. Ens permet dur a terme les funcionalitats principals de compilació i execució dels servidors web.

Per altra banda, he obert un compte al portal web d'Azure de Microsoft, on he creat un servidor per allotjar el servei (API). També eh creat un servei de DB que és Azure Cosmos DB per a la gestió de la base de dades.

En la següent imatge podem fer-nos una idea general:



18. Arquitectura final del back-end.

La situació ideal seria replicar aquest servei en més d'un servidor i tenir un balancejador de carga que anés repartint les peticions depenent la carga dels servidors així com habilitant l'opció de multi-master de Azure Cosmos DB, per a tenir una base de dades per a cada servidor.

Tot aixó, ara per ara es innecessari, però s'ha tingut en compte a l'hora de crear el disseny ja que permetria servir a un gran volum de usuaris si l'aplicació te èxit.

Un dels canvis més important per a que l'escalabilitat sigui el més fàcil possible es en l'acreditació i accés a la API que s'ha realitzat mitjançant tokens, JWT, en ves de per sessions com es típic, ja que el servidor no ha de mantenir ninguna dada del usuari. Però, tot no es bo, per guanyar aquesta escalabilitat perdem en eficiència, ja que transmetim més informació en la comunicació entre la Api i el client(App) i entre la Api i la Base de dades.

S'ha escollit Azure Cosmos DB com ja s'ha comentat anteriorment per la escalabilitat que ens ofereix MongoDB i per la compatibilitat al ser igual que el servidor un servei de Microsoft Azure.

## 10.2 Disseny *back-end*

Donat que he construït un sistema amb un sol servei, una API que gestiona totes les peticions que rep per part dels clients. Cal recordar que la nostra API esta pujada en la URI "<https://apirestenergysave.azurewebsites.net/>" i per a més seguretat qualsevol consulta no valida retornara un *unauthorized*, per a no donar més pistes als possibles atacants.

Anem a enumerar quines són aquestes funcions (*endpoints*):

- **ruta base + api/token:**

Aquest *endpoint* serveix per generar *tokens* d'autorització per als usuaris amb els quals podran accedir a diferents recursos, depenent dels privilegis que tingui el seu usuari. Es l'*endpoint* que es crida en el inici de sessió de l'aplicació.

Es tracta d'una petició post amb dos camps com a cos de la petició, un username i una contrasenya que després de ser validats en la base de dades (Taula/col·lecció Usuaris), si són correctes, com hem dit abans retornara un token amb els camps de *username*, *hotel* i *room* en el seu interior per a futures comprovacions.

\*Com estem parlant de una base de dades de MongoDB deixarem de referir-nos com a Taula/col·lecció i ho farem únicament com a col·lecció, ja que es diu així en MongoDB.

La col·lecció "Usuaris" te la següent estructura:

Users
+ id: Integer
+ username: String
+ password: String
+ hotel: String
+ room: String

- **ruta base + api/hotel/{name}:**

Aquest *endpoint* serveix per a consultar les dades d'un hotel amb el nom "name" rebut per la URL i és un recurs privat al qual sol tindrà accés el administrador de l'hotel.

Per a que la crida sigui del tot segura es comprova que el hotel demanat per la URL sigui el mateix que es rep en el token i que es tracta del usuari administrador.

Es tracta d'una petició get amb una capçalera d'autenticació que contindrà el token de l'*endpoint* d'autenticació.

La col·lecció “Hotels” te la següent estructura:

Hotels
+ id: Integer
+ hotelName: String
+ direction: String
+ city: String
+ directionNumber: Integer
+ rooms: Integer
+ floors: Integer<String>
+ discounts: List<String>

- **ruta base + api/room/{name}:**

Aquest *endpoint* serveix per a consultar les dades d'una habitació amb el nom “name” rebut per la URL i és un recurs privat al qual sol tindrà accés el administrador de l'hotel i l'usuari(client) de l'habitació.

Per a que la crida sigui del tot segura es comprova que l'habitació demanada per la URL sigui la mateixa que es rep en el token o que es tracta del usuari administrador del mateix hotel rebut per token i que té aquesta habitació.

Es tracta d'una petició get amb una capçalera d'autenticació que contindrà el token de l'*endpoint* d'autenticació.

Però abans de seguir, anem a definir l'estructura de dos objectes:

Tip
+ name: String
+ title: String
+ description: String

Period
+ iniDate: String
+ endDate: String
+ efi: Integer

La col·lecció “Rooms” te la següent estructura:

<b>Rooms</b>
+ id: Integer
+ room: String
+ hotel: String
+ floor: Integer
+ door: Integer
+ periods: List<Period>
+ tips: List<Tip>
+ currentPeriod: String
+ currentPeriods: List<Period>

I per finalitzar, les crides més importants, les crides que gestionen les dades importants del projecte:

\*Per a totes les següents crides, Necada envia el nom del hotel, el nom de l’habitació i un *token* en la capçalera, es a dir, prèviament s’ha autenticat.

Cada una d’aquestes crides busca en la Taula “NecadaModels” si ja hi ha la instància amb el seu hotel i habitació i la modifica o sinó la crea.

- **ruta base + api/necada/cloud:**

Aquest *endpoint* serveix per a que Necada actualitzi el nuvol de punts de la base de dades(atribut model).

Necada envia dos XML, un amb els diferents resultats obtinguts de la simulació de totes les possibles combinacions dels paràmetres d’entrada que tè l’hotel i la habitació. I un altre amb els diferents paràmetres resultants(outputs) per a cada una de les simulacions.

És llegeixen i es guarden en l’atribut “model” de la Taula “NecadaModels”.

És guarden en objectes anomenats “Points”:

Point
+ id: Integer
+ inputs: List<Var>
+ outputs: List<Var>
+ simName: String

On els objectes anomenats “Var” són:

Var
+ id: Integer
+ name: String
+ value: Double

- **ruta base + api/necada/anSenOutputs:**

Aquest *endpoint* serveix per a que Necada actualitzi el anàlisi de sensibilitat dels outputs(resultats).

Necada envia la llista de tipus de Outputs del model de simulació amb un *double* assignat a cada un, que serà el pes que aquesta variable té a la hora de calcular l’eficiència que se’ls hi dona als recursos.

Es guarda en l’atribut “anSenOutputs” de la Taula “NecadaModels”.

- **ruta base + api/necada/anSenInputs:**

Aquest *endpoint* serveix per a que Necada actualitzi el anàlisi de sensibilitat dels inputs.

Necada envia la llista de tipus de Inputs del model de simulació amb un *double* assignat a cada un, que serà el pes que aquesta variable té a la hora de aconsellar modificar un input per moure'ns de punt en punt cap al punt optim.

Es guarda en l’atribut “anSenInputs” de la Taula “NecadaModels”.

- **ruta base + api/necada/optimalPoint**

Aquest *endpoint* serveix per a que Necada actualitzi el model optim.

Necada envia un XML amb els resultats del punt optim amb el nom d'aquesta simulació.

Es guarda en l'atribut "optimalPoint" de la Taula "NecadaModels" després de buscar els inputs relacionats.

La col·lecció "NecadaModels" te la següent estructura:

<b>NecadaModels</b>
+ id: Integer
+ hotel: String
+ room: String
+ anSenOutputs: List <Var>
+ anSenInputs: List <String>
+ model: List<Point>
+ optimalPoint: Point
+ currentPoint: Point

- **ruta base + api/necada/tips**

Aquest *endpoint* serveix per a que Necada actualitzi els consells que l'aplicació dona.

Necada envia una llista amb el nom dels inputs del model de simulació associat a consells que se'ls hi dona als usuaris per a modificar aquesta variable.

Es guarda en la Taula "Tips":

<b>Tips</b>
+ id: Integer
+ name: String
+ title: String
+ description: String



- **ruta base + api/necada/currentPoint**

Aquest *endpoint* serveix per a que Necada actualitzi el punt actual en que ens trobem.

Necada envia un XML amb els resultats del punt actual amb el nom d'aquesta simulació.

Es guarda en l'atribut "currentPoint" de la Taula "NecadaModels" després de buscar els inputs relacionats.

També cal recordar els *endpoints* de canvi de contrasenyes, tant de administrador com de clients i les del canvi dels descomptes que són molt simples per tant ja no les mencionarem.

### 10.3. Implementació back-end

Un cop definides les crides importants a la API i les col·leccions que té la base de dades anem a centrar-nos en l'important.

Un cop el comença un període, aquest guarda una data d'inici amb una eficiència.

La eficiència actual es calcula de forma:

$var1\_punt\_optim/var1\_punt\_actual * pes\_var1 + \dots + varN+1\_punt\_optim/varN+1\_punt\_actual * pes\_varN+1 = \text{eficiència total del període}$

on n son el numero de variables resultants que te un punt i els pesos de les variables estan en la llista de sensibilitat dels outputs.

Fins que aquesta eficiència no canvia, Necada llegira que els valors dels inputs(sensors) han canviat no es dona el sub-període per finalitzat. En aquest punt en comença un altre. I mica a mica es van guardant sub-períodes en el atribut de "currentPeriods" de la taula de "rooms".

Un cop finalitza el període, es fa un calcul molt simple que és:

$eficiencia1 * numeroDeMinutsAmbEficiencia1 + \dots + eficienciaN+1 * numeroDeMinutsAmbEficienciaN+1 / numeroDeMinutsTotalsDelPeriode$

on n son el numero de sub-períodes que ha tingut el període, guardats en el atribut de "currentPeriods" de la taula de "rooms".

Però ens deixem la part més important, cada cop que canviem de sub-període hem de servir uns consells al clients per tal de que canviïn a un punt amb una eficiència dels recursos més bona.

Aixó ho fem mirant el punt actual en el que ens trobem, el comparem amb el optim i comencem a avaluar quins inputs del nostre punt canviem per tal de anar variant cap al punt optim.

Es moment de mirar el anàlisis de sensibilitat dels inputs, i comencem a variar els valors per el input que tingui més pes.

## 11. CONCLUSIONS

Amb el projecte acabat, és hora de valorar l'assoliment dels objectius esmentats al principi de la planificació del projecte i extreure les conclusions en l'àmbit personal i professional.

Els objectius principals d'aquest projecte s'han complert tots menys el de implementar notificacions, ja que al ser una aplicació on els usuaris comparteixen comptes, es necessitaria implementar un bon sistema per tal de saber si el usuari esta usant l'habitació i li hem d'enviar una notificació.

Si no es planifica bé, i es pel motiu el qual s'ha descartat fer, crearíem un estimul molest i perdríem als usuaris que tindríem per rebre notificacions d'habitacions en les quals ja no hi són.

Com a treball futur podem dir que caldrà finalitzar per part de Necada la connexió amb l'aplicació, així com instal·lar la infraestructura de sensors amb hotels.

Finalment, podem concloure que la realització d'aquest projecte a set satisfactòria, ja que s'ha realitzat tot el que s'esperava per la part del client en el termini especificat.

## 12. VALORACIÓ PERSONAL

La realització d'aquest projecte ha sigut una bona experiència personal i acadèmica. He aplicat molts coneixements apresos tant en la universitat com en la vida professional.

A més a més, he obtingut coneixements d'un nou llenguatge de programació que no l'havia utilitzat mai, com és c#, après a usar el IDE Xamarin, conèixer i usar tota la infraestructura Azure.

L'avantatge que trobo en no realitzar el TFG en una empresa és que crec que aprens a espavilar a analitzar totes les possibles vies, encara que siguin errònies que en una empresa ja serien descartades, però al fer-ho saps per quin motiu.

En conclusió crec que s'ha fet una bona feina per part del tutor del projecte que actuava al mateix moment com a client i que ha quedat satisfet. A més els conceptes que s'han après són de gran valor.

## Bibliografia

1. Web de NECADA [ Data de consulta: 6 març 2018 ] [En línia].  
Disponible a: <https://necada.com/>
2. Acad de Adobe [ Data de consulta: 18 abril 2018 ] [En línia].  
Disponible a: [https://www.autodesk.es/campaigns/autocad-offers?mktvar002=737894&mkwid=sXQCD1Itj|pcrid|260649978376|pkw|autocad|pmt|p|pdv|c|slid|pgrid|53602991512|ptaid|kwd-28813781|&intent=&utm\\_medium=cpc&utm\\_source=google&utm\\_campaign=GGL\\_AutoCAD\\_UK\\_BR\\_SEM%3EProspecting&utm\\_term=autocad&utm\\_content=sXQCD1Itj|pcrid|260649978376|pkw|autocad|pmt|p|pdv|c|slid|pgrid|53602991512|ptaid|kwd-28813781|&gclid=EAIAIqobChMIysfWxb\\_C2gIVRxDTCh15aQgvEAAYASAAEgLJGPD\\_BwE](https://www.autodesk.es/campaigns/autocad-offers?mktvar002=737894&mkwid=sXQCD1Itj|pcrid|260649978376|pkw|autocad|pmt|p|pdv|c|slid|pgrid|53602991512|ptaid|kwd-28813781|&intent=&utm_medium=cpc&utm_source=google&utm_campaign=GGL_AutoCAD_UK_BR_SEM%3EProspecting&utm_term=autocad&utm_content=sXQCD1Itj|pcrid|260649978376|pkw|autocad|pmt|p|pdv|c|slid|pgrid|53602991512|ptaid|kwd-28813781|&gclid=EAIAIqobChMIysfWxb_C2gIVRxDTCh15aQgvEAAYASAAEgLJGPD_BwE)
3. Web de Eco designer star [ Data de consulta: 6 març 2018 ] [En línia].  
Disponible a: [http://www.graphisoft.com/archicad/ecodesigner\\_star/](http://www.graphisoft.com/archicad/ecodesigner_star/)
4. Projecte de fi de Master [ Data de consulta: 18 abril 2018 ] [En línia].  
Disponible a: <https://upcommons.upc.edu/handle/2117/110281>
5. Web de Xamarin [ Data de consulta: 6 març 2018 ] [En línia].  
Disponible a: <https://www.xamarin.com/>
6. Web de React native [ Data de consulta: 6 març 2018 ] [En línia].  
Disponible a: <http://facebook.github.io/react-native/>
7. Web de Ionic [ Data de consulta: 6 març 2018 ] [En línia].  
Disponible a: <https://ionicframework.com/>
8. Wikipedia. Metodologies Àgil [ Data de consulta: 6 març 2018 ] [En línia].  
Disponible a: [https://es.wikipedia.org/wiki/Desarrollo\\_%C3%A1gil\\_de\\_software](https://es.wikipedia.org/wiki/Desarrollo_%C3%A1gil_de_software)
9. Metodologia Scrum. Guia scrum [ Data de consulta: 6 març 2018 ] [En línia].  
Disponible a: <http://www.scrumguides.org/docs/scrumguide/v1/scrum-guide-es.pdf>
10. Web de Asana [ Data de consulta: 6 març 2018 ] [En línia].  
Disponible a: <https://asana.com/es/>
11. Web de GitHub [ Data de consulta: 6 març 2018 ] [En línia].  
Disponible a: <https://github.com/>
12. Web de Skype [ Data de consulta: 6 març 2018 ] [En línia].  
Disponible a: <https://www.skype.com/es/>
13. Web de Gmail [ Data de consulta: 6 març 2018 ] [En línia].  
Disponible a: <https://mail.google.com>

14. Web de SourceTree [ Data de consulta: 6 març 2018 ] [En línea].  
Disponible a: <https://www.sourcetreeapp.com/>

15. Web de Ganttter [ Data de consulta: 6 març 2018 ] [En línea].  
Disponible a: <https://www.ganttter.com/>

16. Web de Visual Studio [ Data de consulta: 6 març 2018 ] [En línea].  
Disponible a: <https://www.visualstudio.com/>

# Annex – Diagrama de Gantt

