



Escola d'Enginyeria de Telecomunicació
i Aeroespacial de Castelldefels

UNIVERSITAT POLITÈCNICA DE CATALUNYA

TREBALL FINAL DE CARRERA

Títol del TFC: Gestió de recursos per a consum intern d'una organització utilitzant Google Apps Script

Titulació: Enginyeria Tècnica de Telecomunicacions, especialitat Telemàtica.

Autor: Albert Oliva Roura

Director: Jesús Alcober Segura

Data: 15/05/2015

Resum:

La crisi econòmica registrada els darrers anys a nivell global i més especialment la desacceleració del creixement del sector turístic a Espanya, una de les primeres potències mundials al negoci, provoca que els hàbits de consum dels turistes canviïn. Les famílies disposen de menys recursos econòmics per dedicar a les seves vacances i es decanten per fòrmules menys luxoses que els permetin continuar amb els seus costums pel que fa a oci vacacional.

En aquest context el lloguer d'apartaments turístics esdevé per una banda una alternativa més econòmica per poder visitar destinacions amb preus tradicionalment elevats en quant a allotjament, i d'altra banda també constitueix una oportunitat de negoci o com a mínim de rentabilització i manteniment de propietats immobiliàries en un moment en què resulta molt complicat, o directament no recomanable intentar vendre-les o llogar-les de forma permanent. Així, molts propietaris es constitueixen com a llogaters de temporada, evitant els riscos de morositat que comporta el lloguer permanent i no havent de malvendre el seu patrimoni a uns preus de mercat que sovint són fins i tot inferiors a les hipoteques que per ells es deuen.

Estem, doncs, davant d'un model hoteler en alça però encara poc professionalitzat i que té una forta necessitat de reduir els costos de manteniment tant com sigui possible per tal de poder oferir tarifes competitives.

Tenint en compte aquesta realitat, la meva proposta per aquest tipus de negoci consisteix en una eina de gestió d'allotjaments turístics a petita escala, lleugera i totalment portable -fins i tot escalable i personalitzable, si s'escaigués- construïda completament aprofitant els recursos gratuïts de Google, tot coordinant diversos dels seus serveis (Drive, Calendar, Google Apps Script...). Així oferim una solució robusta i senzilla que permet automatitzar el gruix de les tasques de gestió i estalviar temps i recursos, tot oferint una interfície senzilla que pot controlar qualsevol persona sense formació ni coneixements específics.

Title: Resources management for internal use of an organization using Google apps script

Author: Albert Oliva Roura

Director: Jesús Alcober Segura

Date: 15/05/2015

Overview:

Economic crisis occurred in the last years at a global level, and more specifically the deceleration of the touristic sector growth in Spain -one of the world's first powers in the business, makes the tourist's consumption habits change. Families have less economic resources to spend in their holidays and tend to less luxurious formulas that let them go on with their habits regarding to leisure time.

In this context, apartment rental becomes -on one side- a cheaper alternative to visit destinations with traditionally higher accommodation rates, and on the other side it's also a business opportunity or at least profitability and maintenance of real estate properties in a moment when it's very difficult -or even not recommended- for them to try and sell them or hire them on a permanent basis. Therefore many owners constitute themselves as seasonal landlords, avoiding the non-payment risks that permanent rental implies and not having to sell their patrimony at market rates that are often even lower than the propertie's mortgages.

And so we are in front of an emerging accommodation model, still little to no professionalized and that has a strong need of reducing maintenance costs as much as possible so they are able to offer competitive rates.

Taking this reality into account, my proposal for this kind of business consists on a little holiday rental management tool, light and fully portable -even scalable and customizable, if needed- build completely over Google's free resources, coordinating some of its services (Drive, Calendar, Google Apps Script...). This way we can offer a sturdy and simple solution that lets us automate the bulk of our management chores and therefore save time and resources, offering a simple interface that anyone, even without training or specific skills, can control easily.

ÍNDEX

1. INTRODUCCIÓ

1.1 CONTEXT DEL PROJECTE

1.2 REQUERIMENTS DE NEGOCI

1.2.1 Plantejament preliminar

1.2.2 Realitat del mercat: comparativa amb altres opcions

1.2.3 User Stories: fent reals els requeriments a través de les iteracions

1.3 METODOLOGIA EMPRADA

1.4 TECNOLOGIES UTILITZADES

1.4.1 Google Apps Script

1.4.2 Eines de suport

4.2.1 Pivotal Tracker

4.2.2 IDE Google Apps Script

4.2.3 Hangouts

4.2.4 Drive

2. IMPLEMENTACIÓ DEL PROJECTE

PRIMERA RELEASE (31 OCT 2014)

1. ID73882562 Calendaris no actualitzats

2. ID 73882498 Actualització dades restants i calendari

3. ID 75253744 Cancel·lació de la reserva - tatxat

4. ID 73882666 Tot Ok amb el pagament

5. ID 73882702 Avisos que pagament no OK

6. ID 73882828 Color i notificació d'estada de client

7. ID 73883062 El mateix dia de sortida - groc

8. ID 73882612 Setmana abans de l'arribada

SEGONA RELEASE ()

1. ID73882782 Trigger que el pagament és ok

2. ID 82075804 Crear "current bookings" i "past bookings"

3. ID 73882866 Recepció mensual del calendari

4. ID 82075554 Auto reordenació

5. ID 81859730 Reserva cancelada, lletra tatxada

6. ID 81860096 Ressaltar el client que ha pagat tot el que falta

7. ID 73883130 Seguiment després de tot Ok al checklist

8. ID 73883266 Visualització dels calendaris

9. ID 84448414 Tenir en compte que no hi hagi conflicte entre dates

10. ID 81860894 Settings i snippets

TERCERA RELEASE ()

1. ID 73883420 Reporting econòmic

3. AMBIENTALITZACIÓ

4. CONCLUSIONS

4. 1 DEADLINES

5. MANUAL D'USUARI

6. ANNEXOS

6.1 CODI

6.3 MODELS DE MISSATGES DE L'APLICACIÓ

6.2.1 AVISOS AL GESTOR

6.2.2 AVISOS AL PERSONAL DE MANTENIMENT

6.3 BIBLIOGRAFIA CONSULTADA

1. INTRODUCCIÓ

1.1 CONTEXT DEL PROJECTE

Aquesta eina de gestió de lloguer d'apartaments turístics s'ha concebut com una solució a les necessitats de gestió dels llogaters amb un perfil semi professional, amb poques unitats a administrar, sent per tant un perfil d'autònom o molt petita empresa.

La naturalesa de l'aplicació permet que hi hagi més d'una persona connectada a la mateixa, tot i que donat les dimensions de les empreses usuàries no donarà suport a casos de forta concorrència d'usuaris (tot i que eventualment podria abastar-les sense comprometre's la seva estabilitat).

L'objectiu de l'eina és centralitzar i automatitzar les principals tasques que de forma comú es presenten en aquest tipus d'activitat econòmica:

- Les reserves que els clients realitzin a les plataformes o motors de reserves on el llogater hagi contractat un espai pels seus allotjaments es poden canalitzar per diversos serveis gratuïts com ara un CRM, un gestor d'events... o es podria realitzar un script a mida.
- La funcionalitat de Calendari és alimentada automàticament per les noves reserves i els seus eventuais canvis d'estat (cancel·lacions, finalitzacions d'estada...). L'accés al calendari és exclusiu per a l'administrador.
- La funcionalitat de reporting de plannings és una utilitat que envia al personal de manteniment el planning amb totes les estades del mes següent 5 dies abans que aquest comenci, per tal que pugui organitzar-se les tasques de neteja i altres necessitats.
- El report d'events per mail notifica al gestor certs events prèviament configurats, com pot ser el check out d'un client, un impagament, etc.
- La funcionalitat Spreadsheet és un arxiu construït sobre l'eina de fulls de càlcul de Google que cobreix les següents necessitats:
 - Panell de control on s'assenyalen els events i alertes que requereixen acció per part del gestor així com una vista unificada de les reserves actuals i els seus detalls de pagament, client etc.
 - Report econòmic on el gestor té centralitzada la informació referent a pagaments, bestretes per retornar, despeses de manteniment, etc.

- Snippets: Una petita utilitat on es poden informar les traduccions de les paraules claus de la interfície per tal de mostrar l'aplicació en altres idiomes. Aquesta part es pot escalar per afegir nous idiomes amb uns mínims canvis i per tant molt baixa inversió.
- Configuració: en aquest apartat el gestor introduceix les adreces on l'eina envia les notificacions de diversa naturalesa tant a ell mateix com al personal de manteniment.
- Past bookings: un historial de reserves vençudes, bé per cancel·lades abans de produir-se l'estada o per ser estades ja finalitzades i abonades pel client.

1.2 REQUERIMENTS DE NEGOCI

1.2.1 Plantejament preliminar

Per al cas que ens ocupa i amb l'ànim de poder presentar l'aplicació en funcionament en un escenari real, s'ha plantejat el desenvolupament dins un context on el nostre usuari (el gestor o llogater) posseeix 2 pisos, els quals ofereix en lloguer vacacional de forma contínua mitjançant diferents portals web. Els possibles clients poden veure les dates en les que els pisos es troben disponibles per ser llogats i el cost que els suposaria el rang de dates desitjat.

Aquest gestor, haurà de fer-se càrrec de les següents tasques:

- Verificar que els clients paguin el total del cost abans de marxar del pis.
- Gestionar les peticions de servei per que els pisos estiguin nets i en bones condicions.
- Supervisar que no hi hagi solapaments entre estances contractades per diferents clients.
- Portar la comptabilitat del seu negoci.

Per assolir aquests objectius, necessita:

- Funcions d'avisos automatitzades mitjançant la generació de correus electrònics per part de la plataforma.
- Generació de la compabilitat automàtica i al dia.
- Creació i actualització d'un calendari amb totes les ocupacions planejades i passades.

Però, sobretot:

- Que la plataforma sigui prou visual com per que amb un cop d'ull el gestor tingui clar si ha de fer alguna acció sobre un punt crític, ja sigui de tipus morositat d'un client, neteja d'un pis que s'ha deixat, cancel·lació de reserva... la solució ha d'implicar una reducció dels recursos necessaris per administrar el negoci, mai una càrrega de feina extra.

Un cop plantejats aquests requeriments, la primera solució que se'n acut és un CRM o software de gestió de la relació amb els clients, però els problemes que planteja aquesta via són un cost elevat, necessitat d'una infraestructura específica (servidor, hosting, SAI, base de dades...) i el seu corresponent manteniment, tot suposant que no calgui a més fer desenvolupaments a mida.

Un altre inconvenient dels CRM és que són programes difícilment de domini públic, que requereixen d'un aprenentatge previ per fer-los servir.

De manera que la nostra solució hauria de passar per un sistema intuïtiu o fins i tot familiar per a l'usuari típic per tal que des del primer dia s'hi pugui treballar amb facilitat. El nostre sistema a més ha d'enviar requeriments que el gestor pugui entendre, fent servir per tant interfícies i canals de coneixuts pel gran públic i amb els quals el personal administratiu estigui acostumat a treballar. Implícitament, per tant, també ha de ser un sistema que gestioni tant l'entrada de la informació com la seva transformació, organització de mètriques i events i que li doni sortida en un format útil per al gestor.

Com a afegit, i donat que en el nostre cas el llogater té dues propietats a dos ubicacions molt diferents (Bristol i Barcelona situats a 1.178,12 km) es fa imprescindible l'ús d'una eina que ens ajudi a ordenar tota aquesta informació i que a més pugui treballar amb diferents idiomes i divises.

En resum, el sistema ha de comptar amb:

- Disponibilitat online i offline
- Accessibilitat des de qualsevol dispositiu informàtic portàtil o estàtic.
- Multiplataforma.
- Capacitat suficient per suportar la demanda de l'usuari
- Possibilitat de generar còpies de seguretat
- Ser totalment traspassable

1.2.2 Realitat del mercat: comparativa amb altres opcions

A continuació veiem una comparativa amb les característiques de la plataforma que hem creat enfront de tres de les principals opcions que ja són al mercat:

Aplicació	Preu	Connexió amb portals de reserves	Enviment d'avisos	Suport més d'un pis	Control ingressos/despenses	Coordinació personal manteniment	Multi sistema operatiu	Multiplataforma	Cal instal·lar?
La nostra aplicació	0.00 €	SI	SI	SI	SI	SI	SI	SI	NO
Quicken Rental property Manager	154.99€	NO	SI	SI	SI	NO	NO (Win)	NO	SI
Escapia, de HomeAway	4375 €	SI	SI	SI	SI	SI	SI	SI	SI
RNS Vacation Rental Software	€/pis	SI	SI	SI	SI	SI	SI	SI	NO

- **Quicken Rental Property Manager** és una versió del conegut producte Quicken que s'especialitza en el lloguer de propietats inmobiliàries. Com a avantatge sobre el nostre producte podríem dir que és molt potent en la part finançera. Facilita un control exhaustiu de despeses i beneficis, ajuda amb la fiscalitat detectant els elements deduïbles, ajuda a generar previsions de cobrament i despesa, etc. Els inconvenients que fan el nostre més adequat per al model de negoci que ens ocupa són el seu preu i el fet que no és fàcilment integrable amb eines per canalitzar reserves dels portals de viatges.
 - web: <http://www.quicken.com/personal-finance/rental-property-manager-2015>
- **Escapia, de Homeaway**, és un software específic per a gestionar reserves vacacionals. Els seus avantatges enfront de la nostra eina són que pot suportar una quantitat molt gran d'unitats en lloguer, té funcionalitats per crear la pròpia web del negoci, suport tècnic continuat, rols d'usuari amb diferents permisos d'accès, el seu propi sistema de processament de pagaments electrònics, funcionalitat per comandes de servei i manteniment específiques per a cada pis, un mòdul de màrketing per dissenyar publicitat, i un llarg etcètera d'utilitats addicionals. El principal obstacle és el seu preu, més de 4000 euros, que el fa accessible només a negocis de certes dimensions i amb grans beneficis que justifiquin la inversió. Això fa que no sigui una opció aconsellable

per al negoci plantejat, on només hi han dos allotjaments a gestionar i per tant els ingressos i necessitats de gestió són simbòlics.

- web: <http://software.homeaway.com/vacation-rentals/products/escapia>

- **RNS Vacation Rental Software** és una altra solució per a lloguer d'allotjaments, en aquest cas més orientat al cloud. El preu és alt però escalable, ja que es cotitza en cada cas segons el nombre de pisos que es volen gestionar i la modalitat (només cloud, SaaS, software instal.lable)... les seves prestacions són similars a les d'Escapia, afegint alguns extres com poder oferir als clients assegurances de viatge, per exemple. De nou estem davant una proposta molt completa i professional, però que sobrepassa de llarg les necessitats d'un petit negoci amb pocs pisos que no ha de controlar una plantilla de treballadors, proveïdors externs, etc.
 - web: <http://www.rental-network.com/solutions/property-managers/>

1.2.3 User Stories: fent reals els requeriments a través de les iteracions

Com veurem més endavant, hem triat la metodologia Agile per portar endavant el projecte, i això implica que hem transformat cada requeriment en una història d'usuari que s'haurà de fer realitat dins un pla dividit en iteracions. Cada història constitueix una petita comanda de desenvolupament que es transforma a la seva finalització en una nova funcionalitat plenament operativa.

L'estructura d'aquestes històries, que substitueixen els tradicionals documents funcionals, serà la següent:

- Definició: es descriu breument la funcionalitat a implementar. Sovint incorpora uns criteris d'acceptació, que podrien veure's com un "contracte" de les condicions que complirà la història per tal de considerar-se finalitzada (el que es coneix a aquesta metodologia com "definition of done").
- Tasques: la història es trenca en petites tasques de les quals s'en estima el temps a invertir. Les tasques aniran canviant d'estat dins el panell de control del projecte per tal de saber sempre en quin estat tenim el projecte.
- Joc de proves: cada petita història té la seva fase de proves específica per garantir que el programa funciona bé i que el que fa és el que se li demana.
- Pantalles: per tal que el pla de treball acabi constituint una documentació *de facto* de la nostra solució, en acabar cada història hi hem inclòs la imatge de la interface finalitzada.

1.3 METODOLOGIA EMPRADA

Una de les decisions tècniques més importants en encarar un projecte és la metodologia que es vol seguir per portar-lo a terme. Aquesta decisió comporta un compromís amb unes maneres de fer i, per què no, de pensar, i un vot de confiança en un marc de bones pràctiques que creiem que ens portarà a entregar el millor producte possible, de la forma més eficient.

En el cas que ens ocupa ens hem decantat per la metodologia Àgil. El motiu principal per prendre aquesta decisió és la meva experiència professional treballant com a analista de proves de qualitat de software, i la constatació de que altres sistemes de gestió de la producció de programari penalitzen aquest aspecte.

Per exemple, fent una comparació amb el típic mètode de desenvolupament Waterfall o en cascada, el mal endèmic que aquest presenta és que en un inici es pressuposten unes hores específiques per dur a terme cada activitat (presa de requeriments, anàlisi, disseny, desenvolupament, proves, correcció d'errades i més proves per verificar els canvis fets) que després és poc habitual que es compleixin. Encara més, cada petit endarreriment que es registri a les primeres fases de producció penalitza a la última fase, que es va apropiant i després solapant amb la data de lliurament i que per tant sempre es veu retallada. Aquesta retallada en una fase que ha d'assegurar la qualitat no pot presentar res de bo per al client.

Per contra, la metodologia àgil funciona per iteracions en cada una de les quals es projecta acometre un número de funcionalitats prèviament acordat i estimat pel propi equip de desenvolupament, del qual el propi tester n'és un membre important. Aquest ajuda a definir els criteris d'acceptació i porta a terme el disseny i l'execució les proves en el mateix moment en què es desenvolupa la nova funcionalitat. L'impacte de fer les proves abans, i no mesos després de que el programador hagi acabat el codi, implica molts avantatges:

- Feedback immediat del client sobre el que s'està construint
- Detecció precoç de mals plantejaments i bugs
- Com abans es detecti un error, més econòmic és reparar-lo
- Facilitat per adoptar canvis (no s'ha de descartar tant codi ja construït)
- Equips multidisciplinaris: tots els rols prenen part en el producte, augmenta la implicació, la col·laboració entre ells i amb el client
- Possibilitat d'anar fent petits lliuraments periòdics al client amb la corresponent entrada de benefici econòmic, ajudant a sostener el projecte i facilitant el control de costos
- Oportunitat: poder lliurar abans al mercat ens situa més fàcilment com a pioners en el nostre camp i ens dona avantatge sobre la competència

També em va semblar interessant incloure aquesta metodologia al treball de final de carrera perquè tot i que es tracta d'una tendència que augmenta a gran velocitat entre les empreses de desenvolupament de software, sobretot les petites de tipus *start up*, el cert és que aquesta realitat encara no troba en els currículums de la formació universitària del país un reflexe fidel del panorama professional -tot i que ara arrenquen iniciatives de col·laboració amb alguns campus com pot ser l'[Agile Spain University Day](#) celebrat el 2014).

A nivell personal, també trobo que gran part del valor d'aquesta metodologia des del punt de vista del professional rau en que afavoreix la creació de comunitat, el treball en equip i l'intercanvi de bones pràctiques, tot plegat incident en la nostra qualitat com a professionals de les TICs.

Com a conclusió, assenyalar que aquesta forma d'organitzar el treball m'ha permès abordar el TFC amb millor perspectiva, assolint petits objectius de forma continuada en comptes d'enfrontar un camí llarg i farragós més difícil de portar i quantificar, i sobre tot ha permès introduïr canvis i millores de forma constant, sent capaç de lliurar un producte final de ben segur molt millor que el que hagués pogut aconseguir en altres circumstàncies.

1.4 TECNOLOGIES UTILITZADES

Donat la naturalesa lliure del producte que es volia entregar, el més adient i alineat amb l'espiritu de la solució era fer servir recursos gratuïts i disponibles per a tothom i mostrar una forma senzilla però molt eficient d'interconnectar diversos serveis lliures per crear una nova plataforma potent i pensada per a uns propòsits molt concrets, que agrupés una sèrie de serveis i utilitats que d'una altra manera hauríem d'haver cobert acudint a diferent programari, tenint el nostre coneixement de negoci disagregat i dificultant enormement l'explotació de les dades per fer informes, planificacions o tenir visibilitat de la marxa de la nostra activitat econòmica.

1.4.1 Google Apps Script

És un llenguatge d'scripting basat al núvol, propietat de Google i dús lliure i gratuït.

Està basat principalment en Javascript 1.6, tot i que incorpora millores de versions posteriors, i està enriquit amb part de l'API de EcmaScript, un altre llenguatge similar que s'utilitza principalment per a desenvolupaments web a la part client i que és la base d'altres implementacions com ActionScript o el propi JavaScript.

La diferència principal amb aquests és que Google Apps Script s'executa al núvol enlloc de fer-ho al client, amb els avantatges i limitacions que això comporta.

Les principals limitacions són la gairebé absoluta impossibilitat de persistir les dades, les limitacions de temps en quant al temps d'execució dels scripts (Google ho limita per tal que no puguin haver scripts corrent *eternament* al núvol consumint recursos) i la manca d'orientació a objectes, que no afavoreix la mantenibilitat del codi quan l'aplicació comença a tenir certes dimensions i responsabilitats.

Tot i així Google Apps Script ens ofereix la possibilitat d'automatitzar tasques als diferents serveis de Google (Spreadsheets, Mail, Calendar, Drive...) així com connectar-les a serveis de tercers o crear els nostres propis widgets, el qual fa que el potencial de cada una de les seves utilitats augmenti exponencialment.

1.4.2 Eines de suport

Per tal de gestionar el projecte i executar les tasques, s'ha fet ús de diverses eines de suport:

4.2.1 Pivotal Tracker

Es tracta d'una eina de gestió de projectes àgils. És d'accés totalment lliure i gratuït -en cas que es plantegi per fer servir per a un màxim de 3 col·laboradors i 2 projectes privats-, ubicada al núvol, el qual fa que no ens calgui descarregar res i per tant per accedir a les dades del projecte només ens caldrà connexió a internet i qualsevol navegador.

Aquesta aplicació resulta molt útil quan es treballa en equips situats a diferents ubicacions, ja que es té una visió en tot moment del que està fent l'equip i com avança la iteració. A més, en estar específicament enfocada per a la metodologia àgil, ens permetrà configurar certes mètriques característiques de la mateixa com pot ser la velocitat del equip, així com organitzar les tasques en epics, mantenir un backlog de feina pendent de programar, etc. Una eina de gestió de projectes estàndar, tot i hagués pogut arribar a funcionar, hagués estat totalment desconectada d'aquest enfoc de treball i hagués dificultat més que facilitat l'adopció d'aquests principis.

Com a plus, Pivotal té altres utilitats com eines per exportar la informació, la possibilitat d'atorgar diferents nivells de permisos als usuaris sobre les tasques (vista, edició, eliminació, creació), i ens guarda també un històric de les tasques i canvis fets per tenir sempre la possibilitat de traçar les tasques, els temps de compleció, etc.

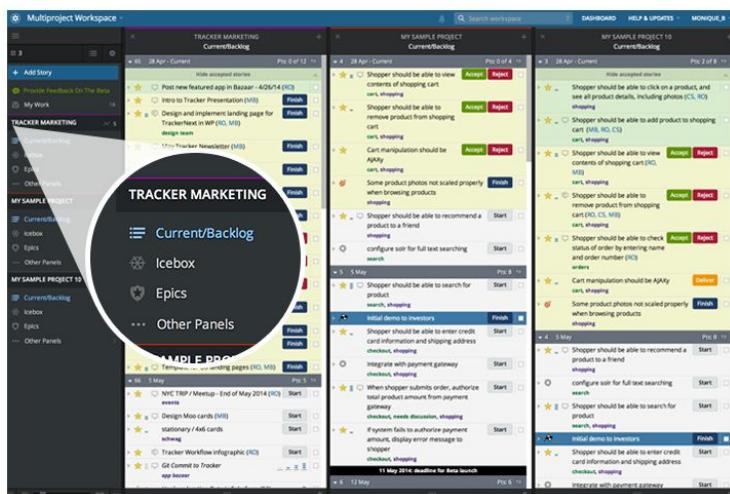


Fig.1 imatge tipus del panell de control d'un projecte àgil gestionat amb Pivotal Tracker.

4.2.2 IDE Google Apps Script

L'IDE que Google posa a disposició per desenvolupar els nostres projectes de Google Apps Script és tan lleuger com una simple finestra de navegador amb un espai on escriure el codi i les principals funcionalitats que podríem trobar a altres editors de codi. Tenim fins i tot la possibilitat de compilar i debuggar des de la mateixa finestra

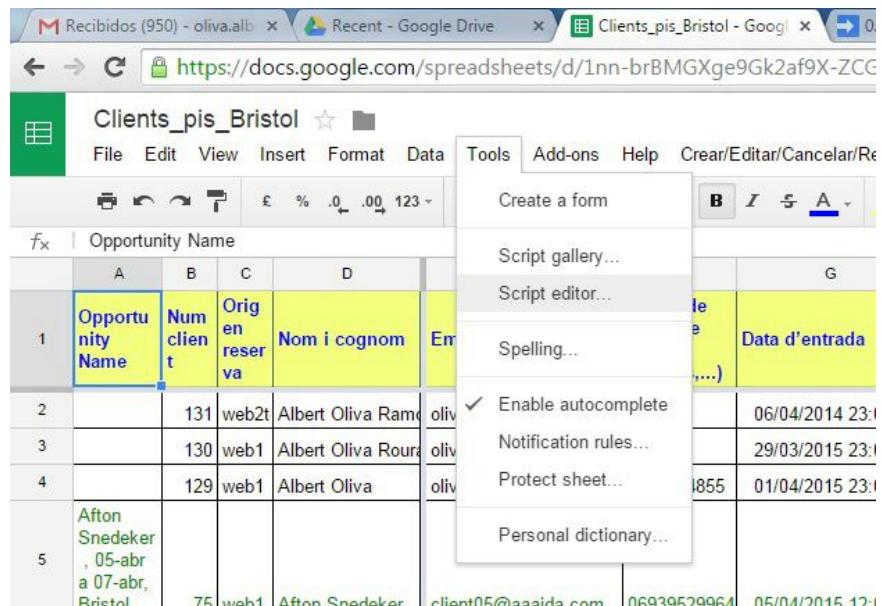


Fig. 2 Punt de menú per accedir a l'editor de codi des del mateix full de dades



Fig.3. Vista típica de l'IDE Google apps script.

4.2.3 Hangouts

És una aplicació de missatgeria instantània desenvolupada per Google que ens permet mantenir conversacions escrites (tipus xat), videoconferències amb xat, o només so amb xat, així com adjuntar-hi arxius de Drive.

Resulta interessant perquè no només permet conversacions d'un a un sino també de tipus broadcast, on una persona emet per a un número indeterminat. Això fa que resulti especialment

interessant per a equips que treballin remotament, encara més si creem grups fixes amb els nostres col·laboradors.

Es tracta també d'una app multiplataforma, que podem fer servir des d'un ordinador només accedint a la seva url amb el navegador -o bé integrat des de Gmail-, o amb mòvils i tauletes descarregant l'aplicació específica.

En aquest projecte Hangouts s'ha fet servir per a mantenir reunions de seguiment sense que calgués desplaçarse i ésser més eficients.

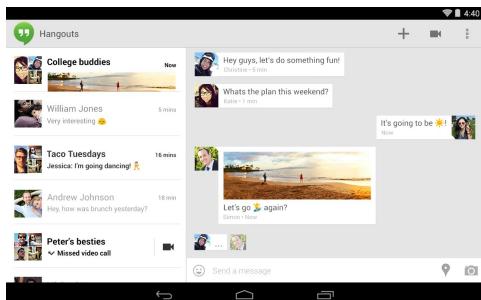


Fig 4. Interface del xat de Google Hangouts amb una conversació oberta, trucades perdudes, etc.

4.2.4 Drive

Drive és el servei d'allotjament d'arxius de Google, basat també en el núvol, tot i que en aquest cas sí que proporciona aplicacions tant per a escriptori com per a dispositius mòbils per tal de millorar l'experiència d'usuari. Per tant, els arxius es poden manipular offline, tot i que sempre caldrà connexió per desar els canvis a Drive si així ho desitgem. Drive accepta arxius adjunts de qualsevol mena, però a més també proposa el seu editor per a certs formats específics: full de càcul, editor de text, editor de presentacions...

Hem fet servir Drive per emmagatzemar el codi font i els arxius relatius al projecte i poder-los compartir amb el tutor.

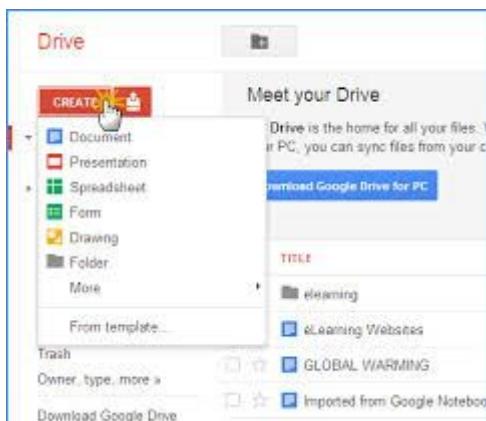


Fig. 5. Vista del panell de control de Drive amb les opcions per crear-hi nous documents.

2. IMPLEMENTACIÓ DEL PROJECTE

Pel nostre projecte comptem amb un full de càlcul ple de files que simulen clients que han reservat el pis. Aquest full conté una serie de columnes que passem a especificar:

1. Opportunity Name
2. Núm client (valor únic) (núm sencer)
3. Origen reserva (text)
4. Nom i cognom client (text)
5. E-mail client (text)
6. Dades de contacte client (telfs, adreces,...) (text)
7. Data d'entrada (data)
8. Data de sortida (data)
9. # dies (núm sencer)
10. # persones (núm sencer)
11. # adults (núm sencer)
12. # nens (núm sencer)
13. País (text)
14. Cost pel client (moneda). Aquest cost és la quantitat total que el client paga per la seva estada. És un valor fix, que se sap des que es fa la reserva
15. Paga i senyal (moneda). Aquesta és la quantitat que el client paga per confirmar la reserva. Es un valor fix, que se sap des que es fa la reserva
16. Saldo (moneda) Aquesta és la quantitat que falta per pagar en la data actual. És la fórmula següent: "Cost pel client" - ("Pagat" + "Paga i senyal")
17. Pagat (moneda) Aquesta és la quantitat que el client ha pagat fins la data actual
18. Despeses de neteja (moneda) Són les despeses establertes de neteja
19. Despeses de gestió (moneda) Són les despeses establertes de gestió
20. Impostos (moneda) Són els impostos associats a l'estada
21. Guanys (moneda) Són els guanys associats a l'estada. És la fórmula següent: "Cost pel client" - "Despeses de neteja" - "Despeses de gestió" - "Impostos"
22. Fiança (tant per cent)
23. Checklist (Booleà)
24. Ressenya (Booleà)
25. Calendaris actualitzats (Booleà): Ens indica si els calendaris de les webs d'on venen les reserves tenen la mateixa informació que el de la nostra aplicació
26. Retorn fiança (Booleà)
27. Comentari (text)
28. Cancel·lacions (Booleà) per saber si una reserva ha estat cancel·lada

29. Gestor avisat (data en que s'envia l'avís) indica si el gestor ha estat avisat mitjançant un correu i evita un bombardeig massiu d'alertes i avisos en un mateix dia.

30. Usuari avisat (Booleà) indica si l'usuari ha estat avisat mitjançant un correu.

Existeixen dos rols en el nostre projecte:

- Gestor del pis:

És l'usuari principal i administrador de l'eina. A ell li pertanyen els pisos i ha de tenir tota la informació disponible i al seu abast.

- Operari de neteja:

Rep notificacions per mirar de fer la seva feina de la manera més eficient possible. Ajudant-se de l'aplicació pot preveure els dies que ha de netejar el pis.

A continuació passem a desglossar les històries d'usuari que componen l'aplicació:

PRIMERA RELEASE (31 OCT 2014)

1. ID73882562 *Calendaris no actualitzats*

Definició:

La fila es pintarà de color groc fins que la columna corresponent a Calendaris actualitzats està a SI, en aquell moment pintarà la fila de color blanc.

CRITERIS D'ACCEPTACIÓ:

- Quan el calendari està sincronitzat correctament, la columna es mostrarà en blanc.
- Quan el calendari està desincronitzat o obsolet, es pinta de groc i es manté així mentre no se soluciona.
- Si per algun motiu el gestor edités el camp o es perdés la info, la columna s'actualitzà automàticament a "NO" per testimoniar que la fulla pot estar no sincronitzada.

Tasques:

- Implementar lògica de colorejat de files
- Implementar la comprovació d'estat dels calendaris

Joc de proves:

1. Efectuar edició a la columna a "FALSE". Verificar que canvia a color groc.
2. Efectuar edició a la columna a "TRUE". Verificar que canvia a blanc.

2. ID 73882498 *Actualització dades restants i calendari*

Definició:

Quan el client potencial contesta positivament, s'actualitza el full de càlcul global amb totes les dades restants (i pinta la fila de color groc) i el calendari global

Tasques:

- Implementar lògica per controlar que les files han estat omplertes amb les dades confirmades del client
- Incrementar la lògica que *decideix* el color de la fila perquè avalui si s'ha rebut nova informació del client

Joc de proves:

- Crear una nova reserva. La cel.la apareix al full global. La fila es pinta de groc al executar l'script si la columna de "Calendaris actualitzats" està a FALSE.
- Crear una nova reserva. La cel.la apareix al full global. La fila es pinta de blanc al executar l'script si la columna de "Calendaris actualitzats" està a TRUE.

3. ID 75253744 Cancel·lació de la reserva - tatxat

Definició:

En qualsevol moment abans de l'arribada el client pot cancel·lar la petició. En aquell moment la fila es queda tatxada, i la fila es pintarà de color groc fins que la columna corresponent a Calendaris actualitzats està a SI, en aquell moment pintarà la fila de color gris.

CRITERIS D'ACCEPTACIÓ:

- Una cancel·lació de reserva provoca que les dades de la fila es tatxin
- La cancel·lació de reserva ha de fer que la dada torni a estar contractable als portals
- Fins que aquesta dada no es propagui als portals de reserves, la columna calendari es mantindrà en groc
- Un cop el canvi es propagui, la fila quedarà en gris el tatxat es manté

Tasques:

- Implementar lògica de pintat de files perquè també actuï a les cancel·lacions de reserva
- Implementar lògica de tatxat de dades a estades cancel·lades

Joc de proves:

- Sobre una reserva creada, informar "TRUE" manualment a la columna de cancel·lacions. Verificar que les dades es tatxen i es pinta de groc. Verificar que la columna "Calendaris actualitzats" està a "FALSE".
- Modificar manualment la columna "Calendaris actualitzats" està a "TRUE". Verificar que el color de la fila canvia de groc a gris i les lletres romanen tatxades.

4. ID 73882666 Tot Ok amb el pagament

Definició:

Una setmana abans de l'arribada del client, si el valor de la columna “Saldo” és 0, llavors envia un missatge al gestor per a que li enviï un missatge de benvinguda al client.

CRITERIS D'ACCEPTACIÓ:

- Si el client ha pagat exactament els diners que sumen el total de la seva estada, la fila es pinta de blanc i s'avisa al gestor per correu electrònic per a que li envii un mail de benvinguda al client

Tasques:

- Lògica de comprovació del valor “Saldo”.
- Incrementar lògica del pintat de cel·les perquè s'activi també per events de balanç de pagament

Joc de proves:

- Donada una reserva amb import 50€, amb data d'entrada d'aquí a una setmana, introduïr manualment 25€ a la columna “pagat”. Verificar que no es rep el correu informant de que s'ha efectuat el pagament total de l'estança.
- Donada la reserva anterior, modificar ara l'import a 50€ i verificar que la fila es pinta de blanc. Verificar que al moment es rep el correu electrònic recordant a donar la benvinguda
- Donada la reserva anterior, modificar ara la data d'entrada per a dos dies després de l'actual. Verificar que la fila es pinta de blanc i que al moment es rep el correu electrònic recordant a donar la benvinguda.

5. ID 73882702 Avisos que pagament no OK

Definició:

Si el valor de la columna “Saldo” és inferior o major que 0, llavors ha d'enviar un missatge al Gestor avisant d'això i pintar la fila de color groc

CRITERIS D'ACCEPTACIÓ:

- Si el client ha pagat més o menys diners que el total de la seva estada, la fila es pinta de groc i s'avisa al gestor per correu electrònic
- Quan la situació es resol (cost pel client = diners pagats) la fila es pinta de verd i s'envia correu al gestor des de 5 dies abans del check in per tal que contacti al client i li doni la benvinguda

Tasques:

- Lògica de comprovació de quantitats que es deuen vs quantitats ja pagades
- Lògica per actualització del balanç de diners pagats
- Incrementar lògica del pintat de cel·les perquè s'activi també per events de balanç de pagament

Joc de proves:

- Donada una reserva amb import 50€, amb data d'entrada d'aquí a dues setmanes, introduïr manualment 25€ a la columna "pagat". Verificar que la fila es pinta de groc i que es rep el correu informant del deute pendent.
- Donada la reserva anterior, modificar ara l'import a 50€ i verificar que la fila es pinta de blanc.
- Donada la reserva anterior, modificar ara la data d'entrada per a dos dies després de l'actual. Verificar que al moment es rep el correu electrònic recordant l'estada.

6. ID 73882828 Color i notificació d'estada de client

Definició:

Durant l'estada del client i sempre que aquest hagi pagat el total de la seva estada, la fila es pinta de color blanc. El mateix dia d'entrada pel matí el gestor rep una notificació.

CRITERIS D'ACCEPTACIÓ:

- Quan la data actual està dins el lapse de data d'entrada - data de sortida, i preu estada = preu pagat, el registre es mostra en color blanc
- Quan la data actual està dins el lapse de data d'entrada - data de sortida, i preu estada > preu pagat, el registre es mostra en color groc

Tasques:

- Incrementar la lògica de pintat de registres perquè contempli aquest nou cas

Joc de proves:

- Donat un full current bookings i un registre amb data d'entrada ahir, i data de sortida demà, i preu = 50€. Introduceix com a import pagat 50€. Verifica que la fila es pinta de blanc.
- Donat un full current bookings i un registre amb data d'entrada ahir, i data de sortida demà, i preu = 50€. Introduceix com a import pagat 25€. Verifica que la fila es pinta de groc. Comprova que es rep el correu avisant del deute pendent.

7. ID 73883062 El mateix dia de sortida - groc

Definició:

El mateix dia de sortida a les 11:00 es pinta la fila de color groc.

Tasques:

- Incrementar la lògica de pintat de files perquè canviï a groc els registres que tinguin data de sortida = dia actual, quan hora actual = 11AM o més tard.

Joc de proves:

- Donada una estada de client amb data d'entrada ahir i data de sortida demà, on import degut = 50€ i import pagat = 50€. Modificar la data de sortida al dia actual. Verificar que la fila es pinta de groc.
- Tornar a modificar la data de sortida a dia actual + 2 dies. Verificar que torna a pintar-se de blanc.

8. ID 73882612 Setmana abans de l'arribada

Definició:

Una setmana abans de l'arribada del client, aquest ha de completar el pagament per tal que no es marqui una incidència (el no pagament no anula el servei).

CRITERIS D'ACCEPTACIÓ:

- Si el client ha pagat exactament els diners que sumen el total de la seva estada, la fila es pinta de blanc i s'avisa al gestor per correu electrònic per a que li envii un e-mail de benvinguda al client

Tasques:

- Lògica de comprovació del valor “Saldo”.
- Incrementar lògica del pintat de cel·les perquè s'activi també per events de balanç de pagament

Joc de proves:

- Donada una reserva amb import 50€, amb data d'entrada d'aquí a una setmana, introduïr manualment 25€ a la columna “pagat”. Verificar que no es rep el correu informant de que s'ha efectuat el pagament total de l'estança.
- Donada la reserva anterior, modificar ara l'import a 50€ i verificar que la fila es pinta de blanc. Verificar que al moment es rep el correu electrònic recordant a donar la benvinguda
- Donada la reserva anterior, modificar ara la data d'entrada per a dos dies després de l'actual. Verificar que la fila es pinta de blanc i que al moment es rep el correu electrònic recordant a donar la benvinguda.

SEGONA RELEASE ()

1. ID73882782 Trigger que el pagament és ok

Definició:

En qualsevol d'aquests moments, si el valor del Cost pel client és igual al valor del camp Pagat, llavors pinta la fila de color blanc i envia un missatge al gestor per a que li enviï un missatge de benvinguda al client.

CRITERIS D'ACCEPTACIÓ:

- Si el client ha pagat exactament els diners que sumen el total de la seva estada, la fila es pinta de blanc i s'avisa al gestor per correu electrònic per a que li envii un e-mail de benvinguda al client

Tasques:

- Lògica de comprovació del valor “Saldo”.
- Incrementar lògica del pintat de cel·les perquè s'activi també per events de balanç de pagament

Joc de proves:

- Donada una reserva amb import 50€, amb data d'entrada d'aquí a una setmana, introduïr manualment 25€ a la columna “pagat”. Verificar que no es rep el correu informant de que s'ha efectuat el pagament total de l'estança.
- Donada la reserva anterior, modificar ara l'import a 50€ i verificar que la fila es pinta de blanc. Verificar que al moment es rep el correu electrònic recordant a donar la benvinguda
- Donada la reserva anterior, modificar ara la data d'entrada per a dos dies després de l'actual. Verificar que la fila es pinta de blanc i que al moment es rep el correu electrònic recordant a donar la benvinguda.

2. ID 82075804 Crear “current bookings” i “past bookings”**Definició:**

L'script genera un nou full al llibre anomenat “past_bookings” que es diferencia de l'actual -on hi han els clients en vigor i els clients futurs- anomenada “current bookings”. El full “past_bookings” no actualitza informació (només servirà per comptabilitat). És a dir, s'enviarà a past (esborrant de current) les files dels clients que ja hagin pagat i hagin sortit de l'habitació i tot estigui OK.

CRITERIS D'ACCEPTACIÓ:

- Si no existeix cap estada cancelada o bé ja pagada i finalitzada a l'històric del negoci, no hi ha full past bookings
- Si existeix una estada que s'hagi cancelat o ja hagi finalitzat i estigui liquidat el seu import, llavors existeix el full “past bookings” i ha de contenir aquest registre.
- Els nous registres s'aniran movent automàticament a past bookings segons es finalitzin i resolguin i apareixeran per defecte en aquest full.

Tasques:

- Crear lògica que implementa un nou full si troba estades cancel·lades o resoltres i finalitzades i l'anomena “past_bookings”
- Crear lògica que filtra els registres susceptibles de ser al full past bookings

Joc de proves:

- Crea un usuari sense cap històric de reserves rebudes. Verifica que no existeix la fulla "past bookings".
- Afegeix una reserva nova i tot seguit informa "SI" a la columna "cancelada". Verifica que es crea el full "past bookings" i que el registre ara apareix allà. Comprova que ja no apareix a current bookings.
- Afegeix una reserva nova ja finalitzada segons dates, amb import a pagar 50€ i import ja satis fet 49€. Verifica que aquest registre no es mou a past bookings.
- Modifica el registre anterior per tal que el valor de la columna checklist sigui "FALSE". Verifica que aquest registre no es mou a past bookings.
- Modifica el registre anterior per tal que el valor de la columna checklist sigui "TRUE". Verifica que aquest registre no es mou a past bookings.
- Modifica el registre anterior per tal que i figuri que l'import pagat és 50€. Verifica que la fulla s'actualitza i el registre ara apareix a "past_bookings" i ja NO figura a "current bookings".

*3. ID 73882866 Recepció mensual del calendari***Definició:**

L'operari de neteja rep el calendari d'entrades i sortides del pròxim mes 5 dies abans de que aquest comenci, per poder-se organitzar. És una còpia d'algunes columnes (de la columna 3 a la columna 9) del full de càlcul de clients.

Tasques:

- configurar els reports per correu electrònic
- configurar les adreces del personal de servei
- disseny i implementació del planning que l'operari ha de rebre

Joc de proves:

- Donat un full amb una sèrie d'entrades futures per al mes següent al vigent, modificar a mà la dada "dia d'avui" per tal que sigui 5 dies abans de l'inici del pròxim mes. Verificar que s'envia el correu amb el planning. Comprovar que el planning es mostra correctament a diferents navegadors i que conté totes les dades.
- Donat el cas anterior, canviar la data de sortida d'una de les futures estades. Verificar que s'envia de nou el planning amb els canvis.

*4. ID 82075554 Auto reordenació***Definició:**

Les entrades dels clients al full s' ordenen per data d'entrada de forma automàtica cada cop que s'executa l'script.

Tasques:

- Implementar la regla d'ordenació per als registres

Joc de proves:

- Donat un full de current bookings amb n registres, comprovar que en executar l'escript aquests apareixen ordenats per data d'entrada.
- Situats a la taula current bookings, modificar la data d'entrada d'un dels registres i comprovar que es re situa al punt de la taula corresponent segons l'esmentada ordenació.
- Donat un full de past bookings amb n registres, comprovar que en executar la solució aquests apareixen ordenats per data d'entrada.
- Situats a la taula past bookings, modificar la data d'entrada d'un dels registres i comprovar que es re situa al punt de la taula corresponent segons l'esmentada ordenació.

*5. ID 81859730 Reserva cancelada, lletra tatxada***Definició:**

Com a gestor vull que si es cancel·la una reserva, quedi sempre amb la lletra tatxada.

Tasques:

- Implementar lògica de visualització de les dades d'un registre d'acord al valor que tingui el concepte "Cancel·lació" (true or false).

Joc de proves:

- Donat un full current bookings amb una reserva que tingui data d'entrada avui + 2 dies i data de sortida avui + 3 dies, amb import 25€. Informar "TRUE" a la columna "Cancel·lacions". Verificar que la lletra ara apareix tatxada.
- Donat el cas anterior, tornar a informar "NO" al camp corresponent de la columna "cancel·lacions". Comprovar que desapareix el tatxat.

*6. ID 81860096 Ressaltar el client que ha pagat tot el que falta***Definició:**

Com a gestor vull que si el client que està actualment allotjat, ha pagat tot el que falta i a més el calendari està sincronitzat, llavors el color de la lletra de la columna sigui diferent, per exemple un verd fosc.

CRITERIS D'ACCEPTACIÓ:

- Sempre que en una estada que s'està produint en aquest moment, tenim totes les dades del client, aquest ha pagat tot i el calendari està sincronitzat, llavors la lletra s'ha de visualitzar en verd fosc.

Tasques:

- Incrementar lògica pintat de files per tenir en compte el cas client allotjat i saldat.

Joc de proves:

- Donada una reserva amb import 50€, amb data d'entrada d'aquí a una setmana, introduïr manualment 25€ a la columna "pagat". Verificar que no es canvia el color de lletra.
- Donada la reserva anterior, modificar ara l'import a 50€ i verificar que es canvia el color de lletra.
- Donada la reserva anterior, modificar ara la data d'entrada per a dos dies després de l'actual. Verificar que no es canvia el color de lletra.

*7. ID 73883130 Seguiment després de tot Ok al checklist***Definició:**

Una vegada tot està correcte i el check list fet, s'ha d'enviar un missatge al gestor conforme tot està correcte i la fila la pinta de color taronja (fins que la columna de Ressenya estigui a SI, el camp corresponent de la columna "Retorn fiança" estigui a SI, llavors la posarà en blanc).

CRITERI D'ACCEPTACIÓ:

- Aquelles estades finalitzades amb check list realitzat donen lloc a l'enviament d'un missatge al gestor comunicant que tot està correcte
- Les estades finalitzades i amb check list realitzat romanen a "current_bookings" i amb la fila de color taronja fins que els camps "retorn fiança" i "ressenya" tinguin valor "TRUE".

Tasques:

- Dissenyar el nou missatge per a aquest cas concret i implementar la lògica i configuració per al seu enviament
- Incrementar la lògica de pintat de files per tenir en compte les estades finalitzades pendents de tornar fiança i ressenya

Joc de proves:

- Donat un full current bookings amb una estada amb data d'entrada ahir i data de sortida dia actual, realitzar check list. Comprovar que es genera el missatge i que el text i aspecte del mateix són els correctes.
- Donat un full current bookings amb una estada amb d'ata d'entrada ahir i data de sortida dia actual, realitzar check list. Informar amb valor "FALSE" els camps "ressenya" i "retorn fiança". Comprovar que la fila es pinta de taronja.

- Donat un full current bookings amb una estada amb data d'entrada ahir i data de sortida dia actual, realitzar check list. Informar amb valor "FALSE" el camp "ressenya" i "TRUE" el camp "retorn fiança". Comprovar que la fila es pinta de taronja.
- Donat un full current bookings amb una estada amb data d'entrada ahir i data de sortida dia actual, realitzar check list. Informar amb valor "TRUE" el camp "ressenya" i amb "FALSE" el camp "retorn fiança". Comprovar que la fila es pinta de taronja.
- Donat un full current bookings amb una estada amb data d'entrada ahir i data de sortida dia actual, realitzar check list. Informar amb valor "TRUE" els camps "ressenya" i "retorn fiança". Comprovar que la ressenya desapareix i passa a past bookings.

8. ID 73883266 Visualització dels calendaris

Definició:

Per a cada pis cal veure les reserves en forma de calendari dels propers 3 mesos i poder veure el detall de neteges i cobraments.

CRITERIS D'ACCEPTACIÓ:

- Els gestors amb més d'un allotjament hauran de tenir un calendari per cada pis que mostri dades a 3 mesos vista, incloent el desglossament del detall de neteges i moviments econòmics.

Tasques:

- Implementar lògica 1 pis = 1 calendari
- Implementar lògica perquè un calendari mostri estades futures amb un límit de dia d'avui + 3 mesos
-

Joc de proves:

- Donat un gestor amb dos pisos (pis 1, Bcn, pis 2, Bristol) on només el pis 1 ha tingut moviment, verificar que només es té un calendari.
- Crear una reserva per al segon pis amb data d'entrada avui + 6 mesos. Verificar que no apareix.
- Crear una altra reserva per al pis 2 amb data d'entrada avui + 1 mes. Verificar que es genera el calendari del pis 2 i que inclou només la segona reserva creada, amb el detall de dades requerit.

9. ID 84448414 Tenir en compte que no hi hagi conflicte entre dates

Definició:

En introduïr una data, s'avisarà al gestor si hi han conflictes amb les dates introduïdes mitjançant un missatge que li demanarà revisar la data, ja que hi deu haver un error.

Tasques:

- Implementar lògica revisió dates en conflicte
- Implementar avís dates conflictives en forma de mail

Joc de proves:

- Donada una reserva preexistente, provar de crear-ne una altra per al mateix pis i amb la mateixa data d'entrada. Verificar que el mail s'envia que es visualitza correctament en els principals navegadors.
- Solucionar el conflicte. Comprovar que ja no es produeix l'avís.

*10. ID 81860894 Settings i snippets***Definició:**

Fer una sheet per settings i una altra per snippets. En els settings permetrà configurar paràmetres dels pisos, per exemple quin es el correu del gestor o gestors, i quin és el correu del personal de neteja. En els snippets permet assignar a una variable el valor dependent de l'idioma, així com també textos per als correus de notificació.

Tasques:

- Crear full settings i configurar els camps i la seva connexió amb les lògiques d'enviament d'avisos i plannings
- Crear full de settings i crear al codi les noves variables
- Informar els literals, és a dir el valor de cada variable a cada idioma suportat
- Crear lògica idioma default i idiomes addicionals que es podran seleccionar des d'un menú

Joc de proves:

- Efectuar test de regressió per verificar que els literals es mostren correctament. Es repetirà l'execució dels test cases vistos fins aquest moment per tal de verificar que totes les funcionalitats i vistes estan mostrant les etiquetes en l'idioma corresponent.
- Al full de settings, modificar els correus electrònics del personal. Verificar que es continuen enviant correctament als nous comptes.

TERCERA RELEASE ()*1. ID 73883420 Reporting econòmic***Definició:**

Pel reporting, tenir en un altre full el total per mes dels tipus: ingressos i despeses dels últims mesos, dels últims anys i dels últims 12 mesos.

CRITERIS D'ACCEPTACIÓ:

- Un report per pis
- El report mostra ingressos, despeses i guanys
- Diferents agrupacions temporals: n mesos, n anys, últim any

Tasques:

- Disseny i creació del nou full
- Lògica un full per pis
- Lògica filtrat i explotació de dades econòmiques

Joc de proves:

- Donat un full de report d'un dels pisos, realitzar les següents verificacions:
 - Les quantitats mostrades al report estan sumades correctament
 - Les quantitats es corresponen amb les que trobem indicades alsfulls de reserves
 - Les reserves futures també apareixen contemplades al report
 - No falta al report cap reserva de les que figuren alsfulls
 - Si editem els valors econòmics d'una reserva, en regenerar el report els imports s'actualitzen

3. AMBIENTALITZACIÓ

Donat que la nostra aplicació s'executa al núvol de Google, ens dóna una possibilitat de multi plataforma que no ens obliga a comprar un equip en concret per desenvolupar les nostres tasques, sinò que amb els dispositius mòvils que ja podem tenir tant a nivell personal com a nivell professional, podem gestionar tants pisos com volguem. No cal fer una inversió en hardware. Això implica que evitem els residus que es produirien en cas que necessitéssim renovar el parc informàtic del nostre negoci, de la mateixa manera que ens estalviem el consum energètic i de recursos que sí que es produiria en la fabricació, transport, venda i gestió d'un hipotètic nou equip.

D'altra banda, en tenir les alertes automàtiques, podem saber si un client ha fet els pagaments corresponents o si un empleat de la neteja està al dia del seu calendari sense haver de fer ús activament d'altres mitjans (com trucades o desplaçaments). Per tant, es produeix un estalvi energètic i es limiten les emissions de CO₂ que es produirien si necessitesim anar a visitar l'operari, el client etc.

Per últim, el fet de tenir la informació sempre disponible, poder-la enviar, exportar o guardar en local, ens liura de la necessitat de traduir totes aquestes dades al paper. L'estalvi econòmic i ecològic d'aquesta manera de treballar té també un fort impacte al medi ambient.

4. CONCLUSIONS

Un cop desenvolupada la solució i superats els obstacles tècnics trobats, aquestes són les conclusions obtingudes:

- S'ha assolit l'objectiu de fer una eina lleugera, econòmica i ràpida que cobreixi les principals necessitats que pot tenir un client que no ha de gestionar gaires unitats ni serveis de tercers. El nostre programa permet a l'usuari tenir sempre una visió clara i concisa de l'estat del negoci, la disponibilitat, les tasques pendents i les completades, i a més li permet accedir al l'historic sempre que ho necessiti.

Com a obstacles trobats durant la implementació, cal dir que bàsicament són els propis de les limitacions que Google imposa al seu entorn i llenguatge:

- No poder orientar a objectes complica molt la tasca de crear un codi lleigible i escalable que a més tingui responsabilitats molt diferents (crear, modificar, accedir i destruir elements) sobre classes i objectes de diferent naturalesa
- Les limitacions i quotes diàries de Google en quant a temps d'execució, transferència de kb, crides a base de dades, etc. que obliguen a optimitzar una eina que no podrà excedir cert número de crides, kb's, pes de dades en mails... ja que si això succeeix el programa llença excepcions i atura l'execució de l'script immediatament. En aquest enllaç se'n poden consultar els termes: <https://developers.google.com/apps-script/guides/services/quotas>
- No poder fer servir la persistència de dades, almenys no de formes ortodoxes. Les variables generades es creen i es destrueixen cada cop que corre l'script, i això complica el punt anterior en quant a dades i temps d'execució.

No obstant tot això finalment s'ha aconseguit una eina estable i portable que es pot accedir des de dispositius mòbils, tauletes o ordinadors amb bon rendiment i una visualització prou adaptada a les diferents pantalles i plataformes, que es pot utilitzar per qualsevol sense formació prèvia, per tant no sols no requereix inversió econòmica si no que la corba d'aprenentatge inicial resulta també negligible.

Com a temes no abordats i que queden per a actualitzacions futures, principalment destacar el **calendari** i el seu potencial. Es poden afegir avisos del tipus sms o fins i tot respondre a una proposta de cita de manera positiva, negativa o neutra i es podria utilitzar l'aplicació en aquest aspecte per poder donar més feedback al gestor del pis amb els seus empleats o proveïdors.

I com a millora final, la protecció dels fulls contra l'edició del codi. S'ha intentat donar al gestor les eines necessàries per a que no hagi de tocar mai la fulla, però si per accident mogués les cel·les de posició, és molt fàcil que l'aplicació deixi de funcionar correctament. Es va investigar com protegir les fulles però un cop protegides no es podien editar de cap manera, la qual cosa no era el que es buscava.

4. 1 DEADLINES

Donat que la fulla no es pot protegir contra edició, és possible canviar format de dades, introduir dades incorrectes per s'script o arribar a desmontar la taula. Una possible millora de cara a un futur es tancar l'accés a l'edició directe de la taula enlloc de mitjançant els formularis.

La traducció completa de l'aplicació és una altre millora ja que actualment estan traduits tots els emails i la majoria de missatges que apareixen per pantalla. Però falten alguns com, per exemple, la capçalera de les columnes de la pàgina principal.

Actualment, amb les dades que genera el resum econòmic, es podrien generar gràfics que donarien la informació de forma més visual. Això es pot automatitzar també i no només això, si no generar gràfics de més tipus (països de més afluència, dates amb més demanda) i d'aquesta manera poder generar campanyes publicitaries més adequades als mercats on poguem tenir més èxit.

5. MANUAL D'USUARI

FUNCIONAMENT GENERAL

Aquesta plataforma de gestió és una solució multi plataforma que té com a motor un script que automatitza les comprovacions i gestions diàries.

Diàriament a les 11 AM **l'script s'autoexecuta** recorrent totes les actuals i futures reserves i pintant de colors aquelles que requereixen l'atenció del gestor de pis. A més, envia e-mails tant al gestor com a l'operari de manteniment, reordena el full enviant les reserves vençudes i liquidades al full Past_bookings.

AGREGACIÓ DE NOUS PISOS

Per utilitzar una fulla excel amb dades d'un nou pis i poder executar el mateix script donem per suposat que el format de la data es dd/mm/aaaa. Les columnes tenen el mateix nom que les fulles dels pisos anteriors i si no cal modificar-les y o afegir-les si no existeixen (l'ordre es pot modificar). Un cop en aquest punt cal seguir aquests pasos:

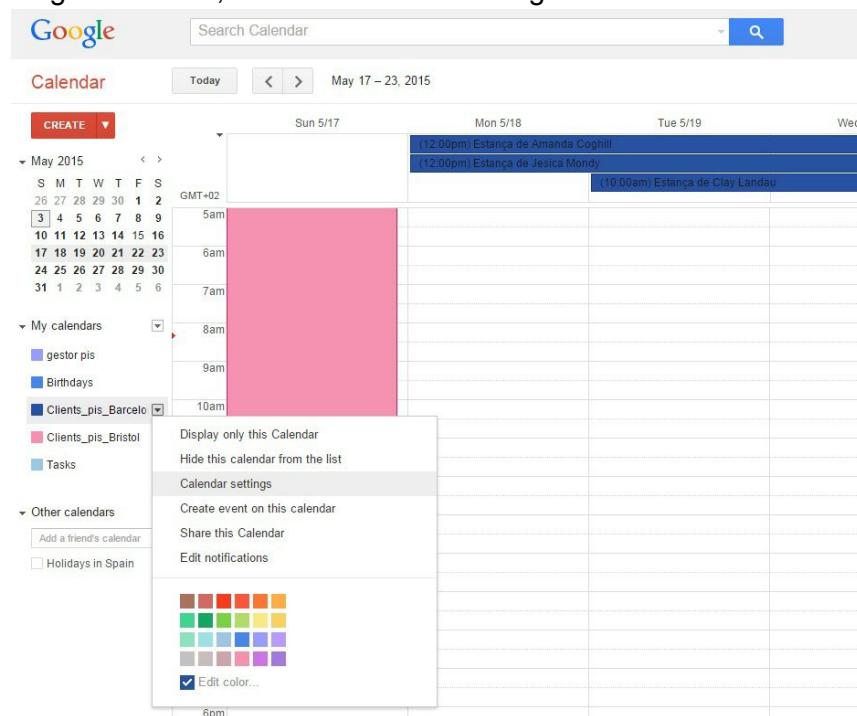
1. Crear pestanya “Settings” crear les cel·les
 - a. “calendarID”. D'aquesta manera s'script crearà un calendari i guardarà l'identificador per poder-hi accedir a la cel·la del costat.
 - b. Manager email: afegir els e-mails del o dels gestors en les cel·les de les columnes següents (accepta columnes buides).
 - c. Cleaner email: afegir els mails del o dels netejadors en les cel·les de les columnes següents (accepta columnes buides).
 - d. Address: Afegir la direcció física del pis a la cel·la de la columna següent.
2. Crear pestaña “Snippets”: cal crear les cel·les per traduir els missatges (és aconsellable fer copy-paste d'una existent, ja que l'script beu directament de les columnes i les files sense fixar-se en el nom de la variable).

3. Agregar l'script: Al full excel anar al menú Tools / Script editor i enganxar i guardar l'script allà. (Si no es veuen els menús adicionals és pq aquests es creen al obrir la fulla, cal tancar i tornar a obrir).
4. Executar l'script per a que s'auto generi la pestanya “Past_bookings” i s'omplia mb les entrades antigues i/o cancel·lades.

ESBORRAR TOT EL CALENDARI D'UN PIS

Si es vol esborrar tot el calendari d'un pis en concret per a que es tornin a generar les entrades en aquest s'han de seguir aquests passos:

1. Go to Google Calendar, Select “Calendar settings”



a.

Fig.6 Gestió del panell de control del Google Calendar: Anar les preferències d'un calendari.

2. Seleccionar “Permanently delete this calendar”

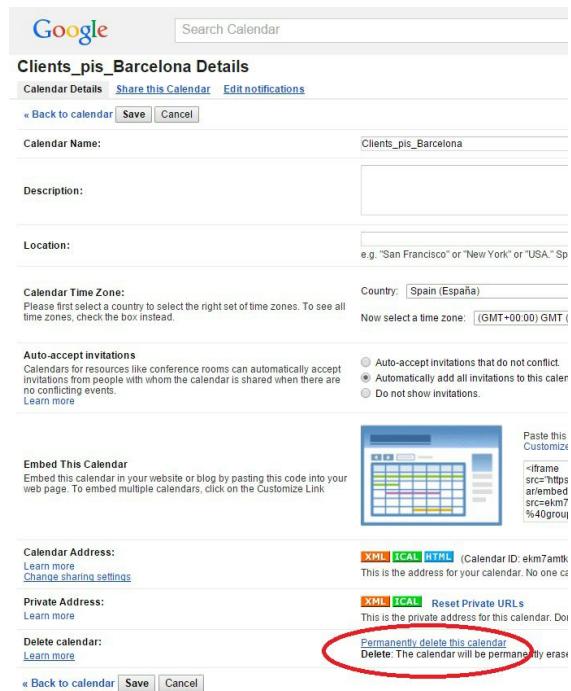


Fig.7 Gestió del panell de control del Google Calendar: Eliminar un calendari.

3. Tria "Delete for everyone"
4. Esborra la ID de la pestanya "Settings" al costat de la cel·la "calendarID".
5. Executa de nou l'script.

CREACIÓ DE NOVES RESERVES

El nostre full de reserves es pot connectar amb els fluxes d'entrada dels motors de reserves de les principals pàgines de viatges mitjançant aplicacions de tercers (el CRM Insightly per exemple), però també podrem alimentar-lo de forma manual mitjançant un formulari en el qual podrem informar totes les dades d'una nova reserva o editar una reserva existent.

Per informar una nova reserva:

- Anirem al menú superior del full de càlcul, anomenat "Crear/Editar/Cancelar/Reporting/Language/Run script" i seleccionarem "Nova entrada" a la opció de menú "Nous".
- Ens apareix un formulari en el qual podem omplir tots els camps de la reserva.
- Un cop estan totes les dades introduïdes, fem clic al botó "Ok" que hi ha al final del formulari
- El full de càlcul es reordenarà segons la data d'entrada de la reserva.

Opportunity Name	Num client	Orig en reserva	Nom i cognom	Email client	Dades de contacte (telfs, adreces,...)	Data de sortida	# dies	# personnes	# adults
1	131	web2t	Albert Oliva Ram	oliva.albert@gmail.com	55	06/04/2014 23:00:00	10/04/2014 23:00:00	2	2
2	130	web1	Albert Oliva Roura	oliva.albert@gmail.com	88	29/03/2015 23:00:00	31/03/2015 23:00:00	2	2
3	129	web1	Albert Oliva	oliva.albert@mail.com	6054884855	01/04/2015 23:00:00	04/04/2015 23:00:00	2	2

Fig.8 Crear una nova entrada a l'aplicació mitjançant el menú.

Nuevo cliente

Oportunity name

Número de client

Origen reserva

Nom i cognom

Email client

Dades de contacte (telfs, adreses,...)

Data d'entrada

Fig.9 Formulari d'entrada de dades per una nova reserva.

Per editar una reserva existent:

- Anirem al menú superior del full de càlcul, anomenat “Crear/Editar/Cancelar/Reporting/Language/Run script” i seleccionarem “Editar” a la opció de menú “Existents”.
- Indicarem el número de client al quadre de text que ens apareix i fem clic a “Ok”

- Es carregarà un formulari en el qual estaran omplerts tots els camps de la reserva del client que hem indicat amb les seves dades per tal que els editem.

Opportunity Name	Num client	Orig en reserva	Nom i cognom	Email client	Dades contacte (telfs adreces)	Data de sortida	# dies	# personnes
1	131	web2t	Albert Oliva Ramí	oliva.albert@gmail.com	55	06/04/2014 23:00:00	10/04/2014 23:00:00	2
2	130	web1	Albert Oliva Roura	oliva.albert@gmail.com	88	29/03/2015 23:00:00	31/03/2015 23:00:00	2
3	129	web1	Albert Oliva	oliva.albert@gmail.com	6054884855	01/04/2015 23:00:00	04/04/2015 23:00:00	2

Fig.10 Editar una entrada a l'aplicació mitjançant el menú.

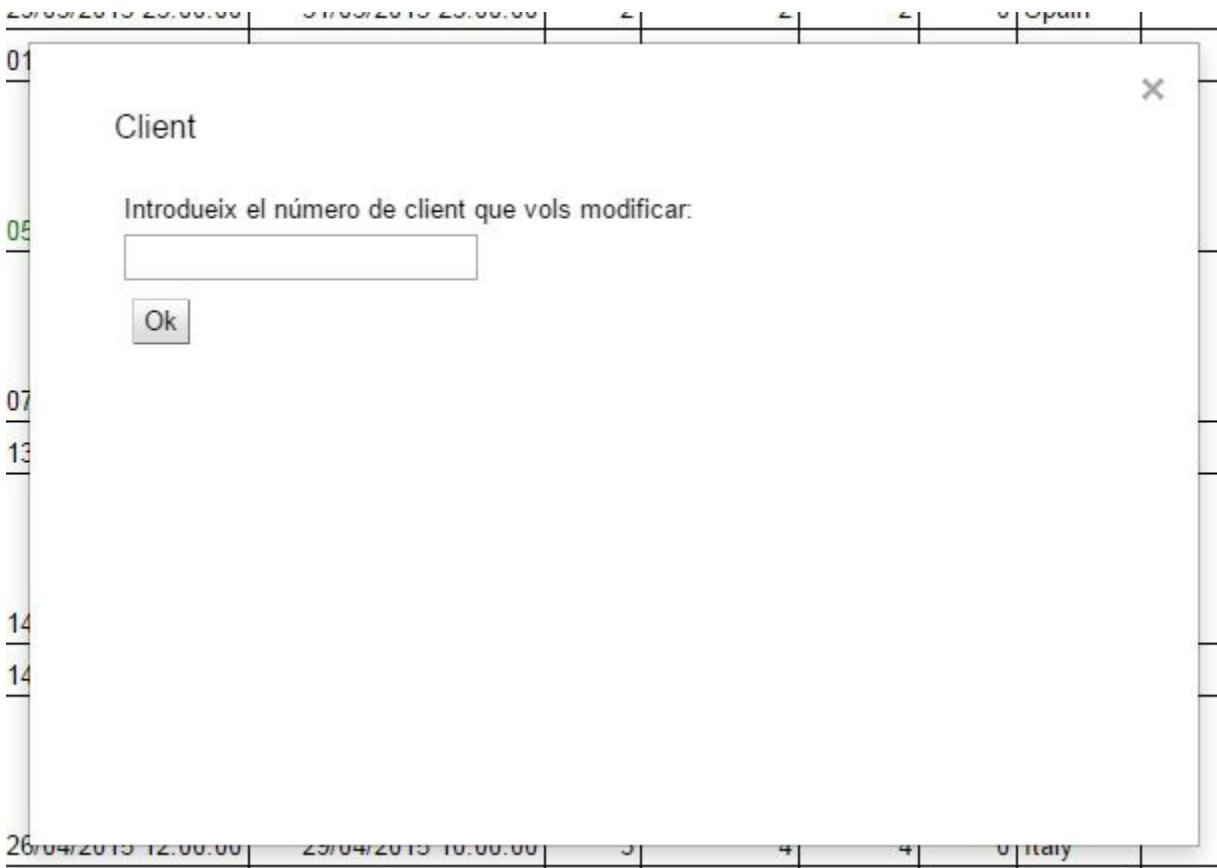


Fig.11 Panell d'introducció del número de reserva el qual es vol editar.

Un cop editat, l'script comprova si la data que hem modificat està compresa en el període del mes vigent, alterant d'aquesta manera el planning del personal de neteja. Per tant, si es dona el cas, s'enviarà un e-mail a l'operari de neteja amb la nova planificació del mes vigent i la reserva

modificada (per si hi han hagut cancel·lacions, ja que en aquest cas no es veuria la fila a la planificació i podria dur a equivocacions).

Independentment, l'script també comprova si el que s'ha modificat és si el pis s'ha pagat i de ser així, envia un e-mail al gestor conforme ja s'ha abonat tot el cost de l'estada.

Per cancel·lar una reserva existent:

- Repetir els passos per a editar una reserva; al formulari trobarem un checkbox anomenat “Cancel·lacions”. Sempre que aquest estigui marcat s'entendrà que la reserva està cancel·lada.
- Marquem el checkbox “Cancel·lacions” i premem “Ok”
- Tornarem a veure la fulla de reserves. L'script s'executarà de forma automàtica i la reserva que hem cancel·lat passarà al full Past_bookings i desapareixerà del calendari de l'allotjament.

England	£300.00	£100.00	£50.00	£40.00	£30.00	£32.00	£13.00	£32.00
Italia								
Polònia								
Polònia								
Espanya								

Fig.12 Panell Edició de l'aplicació.

SERVEI D'AVISOS EN PANTALLA

El programa ens avisarà de que algunes reserves requereixen la nostra acció com a gestors mitjançant un codi de colors amb el qual “pinta” la fila sobre la qual ens vol fer posar la nostra atenció.

El codi de colors és el següent:

BLAU + Lletres de VERD: Es pinten les cel·les de blau i les lletres de verd les reserves corresponents a estades que s'estan produint en el moment en què tenim l'aplicació oberta, i que ja ho tinguin tot pagat.

GROC: Es pinten de groc les reserves que requereixen acció per part del gestor, en dos casos: el client no ha pagat el total de la reserva (per tal que li demanem), o bé el mateix dia de la sortida a les 11h (per tal que es facin les tasques de check out).

GROC + TATXAT: Apareixen així les reserves cancel·lades abans de que la nova disponibilitat es propagui als calendaris i motors de reserves.

GRIS + TATXAT: Quan una reserva es cancel·la, i la nova disponibilitat s'ha actualitzat als calendaris i portals de reserves, llavors la reserva queda en gris i tatxada.

TARONJA: Les reserves ens apareixeran en taronja quan l'estada hagi finalitzat, el check list estigui fet i ens manqui retornar els diners de la fiança i fer review.

SERVEI D'AVISOS PER CORREU ELECTRÒNIC

Els avisos que genera automàticament l'script són:

- **Correus electrònics per al gestor:**
 - **Avís de no pagament.** Quan falten menys de 7 dies per a que arribi un client, si aquest no ha pagat el total de l'allotjament s'envia aquest avís al gestor. Al full de reserves, la fila es pinta de color groc fins que el client pagui el total de l'import de la reserva.
 - **Avís d'arribada de client.** El mateix dia que ha d'arribar un client.
 - **Avís de solapament de dades.** Cada dia es comprova si hi ha alguna reserva que col·lisiona en el temps amb alguna altre. No es gaire probable ja que això està controlat pels motors de recerca però per seguretat, si es detecta alguna col·lisió, s'envia un mail al gestor per a què corregeixi aquesta situació.
- **Correus electrònics per a l'operari de manteniment:**
 - **Avís de la sortida d'un client.** Per tal que aquest pugui fer les tasques de check out que es tinguin planificades.
 - **Resum del calendari de les reserves del mes entrant:** l'operari rebrà un planning mensual amb totes les reserves; li arribarà sempre 5 dies abans que comenci el mes corresponent així com si hi hagués qualsevol modificació posterior, per tal que sempre disposi d'una versió actualitzada.

L'script també s'executa quan hi ha una edició a la columna “Pagat” o a qualsevol columna de les reserves durant el més vigent pero tal d'executar les accions que es poguessin derivar d'aquests canvis.

CALENDARIS DELS ALLOTJAMENTS

Si no existeix prèviament, l'aplicació crea un calendari per cada pis on s'actualitzen les reserves d'aquests. Per tant no es necessari que l'usuari faci cap acció per mantenir o crear els calendaris.

RESUMS ECONÒMICS

Tenim l'opció de generar un report econòmic del període de temps que escollim.

La forma de crear el report és la següent:

1. Accedir al menú “Crear/Editar/Cancelar/Reporting/Language/Run script”
2. Seleccionar la opció “Reporting”
3. Apareix un formulari on haurem d'indicar una data d'inici i de final, d'aquesta manera triem el període del qual volem conèixer les dades
4. Apareixerà el report al full amb la forma d'una una pestanya nova amb dues taules d'ingressos i despeses: una taula amb el detall de cada mes i una altra amb el detall de cada any. El nom de la pestanya serà el del rang de dates que haguem triat, d'aquesta manera si demanessim més d'un report els podríem distingir a primera vista.
5. L'usuari haurà de tenir en compte que, en cas de fer un canvi de dades a un període comprès dins un informe ja generat, serà necessari esborrar la pestanya i regenerar-lo per tal de tenir-lo actualitzat. Es treballa d'aquesta forma per tal que l'usuari tingui sempre consciència del que està veient i per tal que mai pugui tenir obertes dues versions del mateix informe.

A tenir en compte: el resum té en compte l'import que l'usuari ha pagat fins al moment, per tant si ja ha fet algun pagament aquest es comptabilitzarà.

	A	B	C	D	E	F	G	H	I	J
1	Opportunity Name	Num client	Orig en reserva	Nom i cognom	Email client	Dades conta (telfs adreç)	Data de sortida	# dies	# personnes	
2		131	web2t	Albert Oliva Ram	oliva.albert@gmail.com	55	06/04/2014 23:00:00	10/04/2014 23:00:00	2	2
3		130	web1	Albert Oliva Roura	oliva.albert@gmail.com	88	29/03/2015 23:00:00	31/03/2015 23:00:00	2	2
4		129	web1	Albert Oliva	oliva.albert@gmail.com	6054884855	01/04/2015 23:00:00	04/04/2015 23:00:00	2	2
5	Afton Snedeker 05 Jahr									

Fig.13 Punt de menú per accedir a la generació del resum econòmic.

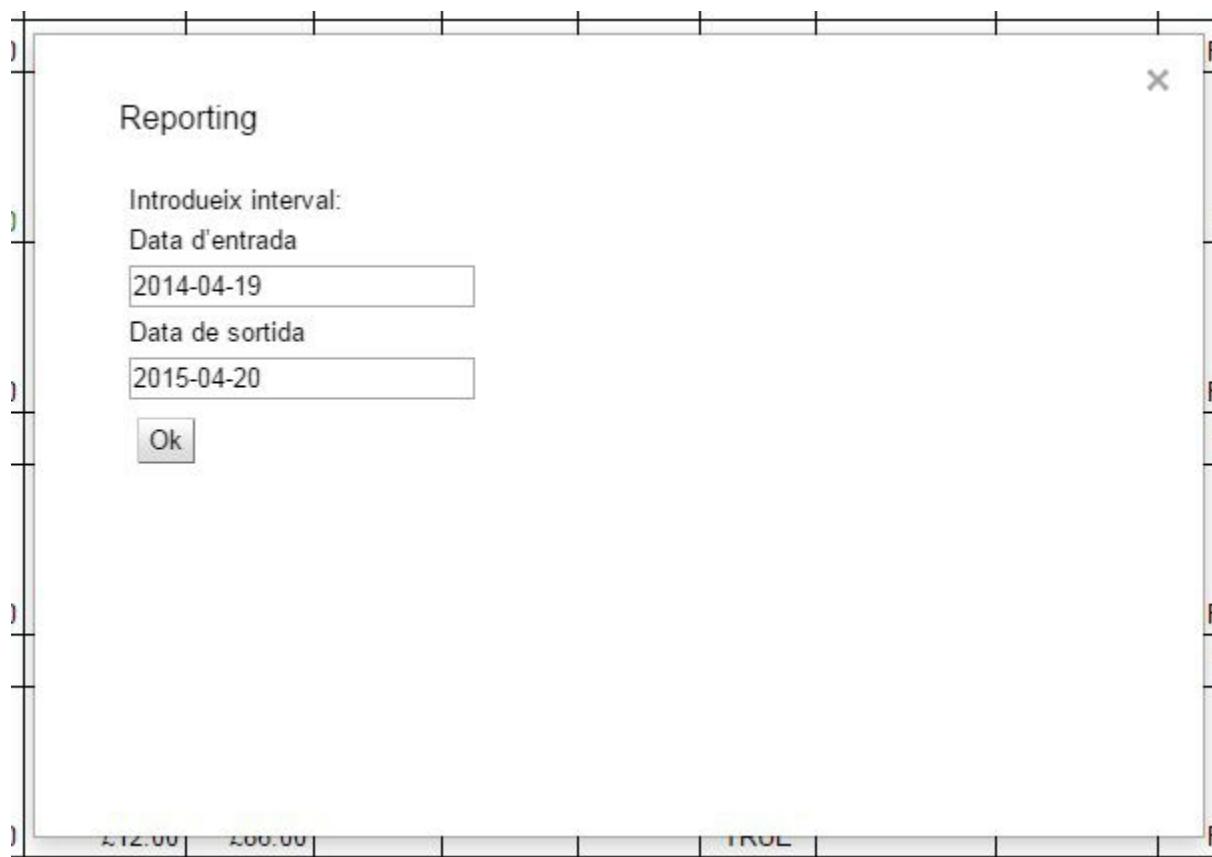


Fig.14 Panell per introduir l'interval de dates en el qual volem que l'aplicació faci el resum econòmic.

	A	B	C	D	E	F	G	H	I	J	K	L	M
1													
2	tipo periodo	ABRIL 2014	MAYO 2014	JUNIO 2014	JULIO 2014	AGOSTO 2014	SEPTIEMBRE 2014	OCTUBRE 2014	NOVIEMBRE 2014	DICIEMBRE 2014	ENERO 2015	FEBRERO 2015	MARZO 2015
3	Ingresos	540	1560	1440	1800	480	1200	1320	360	840	780	720	780
4	Gastos	178.8	422.2	560	603	268	370	383	189	251	255	250	255
5	Beneficio	361.2	1137.8	880	1197	212	830	937	171	589	525	470	525
6													
7	tipo periodo	2014	2015										
8	Ingresos	9540	1690										
9	Gastos	3225	549										
10	Beneficio	6315	1141										
**													

Fig.15 Resum econòmic generat.

6. ANNEXOS

6.1 CODI

Codi de la solució:

```
//Deletes all triggers in the current project.
var triggers = ScriptApp.getProjectTriggers();
```

```

for (var i = 0; i < triggers.length; i++) {
  ScriptApp.deleteTrigger(triggers[i]);
  Utilities.sleep(100); //sleep 0.1 second between deleted trigger
}

// Runs at 11am in the timezone of the script
// Frequency (everyDays) is required if you are using atHour() or nearMinute()
ScriptApp.newTrigger("setRowColors").timeBased().atHour(11).everyDays(1).create();
// Set the trigger to runs on open spreadsheet
ScriptApp.newTrigger("onOpen").forSpreadsheet(SpreadsheetApp.getActiveSpreadsheet()).onOpen().create();

```

Per a que l'aplicació crei els menús addicionals que no incorpora la spreadsheet de google cal crear un trigger en el moment d'obrir el full.

L'altre trigger que veiem és el que fa disparar la funció “setRowColors” (el motor de l'aplicació) cada dia a les 11 del matí ja que és quan es produueixen els checkouts i això pot ocasionar més moviment d'estats als pisos.

Per tal de crear aquests triggers de forma automàtica cal primer comprovar que no existeixin i si existeixen esborrar-los per a que es pugui tornar a crear cada cop que s'executi l'script. Aquest procés pot dur molta càrrega al servidor i per això hi ha un període de temps d'espera forçat entre esborrat de trigger i esborrat de trigger.

```

function setRowColors() {
  // The code below will log the index of a sheet named "Past_bookings"
  var ss = SpreadsheetApp.getActiveSpreadsheet();
  var sheetName = ss.getSheetByName("Past_bookings");
  if (sheetName == null) {
    ss.insertSheet("Past_bookings", 3);
  }
  // The code below will copy from active sheet the first row to another sheet
  ss.getSheets()[0].getRange(1, 1, 1,
    SpreadsheetApp.getActiveSpreadsheet().getSheets()[0].getDataRange().getLastColumn()).copyTo(ss.getSheets()[3].getRange(1, 1, 1,
      SpreadsheetApp.getActiveSpreadsheet().getSheets()[0].getDataRange().getLastColumn()),
    {contentsOnly: false});

  // Freezes the first row
  ss.getSheets()[3].setFrozenRows(1);

  // Freezes the 3 first cols
  ss.getSheets()[3].setFrozenColumns(3);

  // The code below will make the first sheet active in the active spreadsheet
  ss.setActiveSheet(ss.getSheets()[0]);

  var lastColumn = SpreadsheetApp.getActiveSheet().getLastColumn();
}

```

```

var range2 = SpreadsheetApp.getActiveSheet().getRange(1,1,1,lastColumn);

for (var x = 0; x < range2.getLastColumn(); x++) {
  if (range2.offset(0, x, 1, 1).getValue() == "Origen reserva ") {
    var origenCol = x;
  }
  if (range2.offset(0, x, 1, 1).getValue() == "Nom i cognom ") {
    var nameAndSurnameCol = x;
  }
  if (range2.offset(0, x, 1, 1).getValue() == "Email client ") {
    var emailCol = x;
  }
  if (range2.offset(0, x, 1, 1).getValue() == "Dades de contacte (telfs, adreces,...) ") {
    var contactDataCol = x;
  }
  if (range2.offset(0, x, 1, 1).getValue() == "Data d'entrada ") {
    var dateInColumnCol = x;
  }
  if (range2.offset(0, x, 1, 1).getValue() == "Data de sortida ") {
    var dateOutColumnCol = x;
  }
  if (range2.offset(0, x, 1, 1).getValue() == "# dies ") {
    var numDaysCol = x;
  }
  if (range2.offset(0, x, 1, 1).getValue() == "# persones ") {
    var numPersonCol = x;
  }
  if (range2.offset(0, x, 1, 1).getValue() == "# adults ") {
    var numAdultsCol = x;
  }
  if (range2.offset(0, x, 1, 1).getValue() == "# nens ") {
    var numChildCol = x;
  }
  if (range2.offset(0, x, 1, 1).getValue() == "Pais ") {
    var countryCol = x;
  }
  if (range2.offset(0, x, 1, 1).getValue() == "Cost pel client ") {
    var costToTheCustomerCol = x;
  }
  if (range2.offset(0, x, 1, 1).getValue() == "Paga i senyal ") {
    var depositPaymentCol = x;
  }
  if (range2.offset(0, x, 1, 1).getValue() == "Saldo ") {
    var balanceCol = x;
  }
  if (range2.offset(0, x, 1, 1).getValue() == "Pagat ") {
    var paidCol = x;
  }
  if (range2.offset(0, x, 1, 1).getValue() == "Despeses de neteja ") {
    var cleaningExpenditureCol = x;
  }
}

```

```

if (range2.offset(0, x, 1, 1).getValue() == "Despeses de gestió ") {
  var managementExpenditureCol = x;
}
if (range2.offset(0, x, 1, 1).getValue() == "Impostos ") {
  var managementTaxCol = x;
}
if (range2.offset(0, x, 1, 1).getValue() == "Guanys ") {
  var managementGainCol = x;
}
if (range2.offset(0, x, 1, 1).getValue() == "Fiança ") {
  var managementFinancesCol = x;
}
if (range2.offset(0, x, 1, 1).getValue() == "Checklist ") {
  var managementChecklistCol = x;
}
if (range2.offset(0, x, 1, 1).getValue() == "Ressenya ") {
  var managementReviewCol = x;
}
if (range2.offset(0, x, 1, 1).getValue() == "Calendaris actualitzats ") {
  var calendarUpdatedCol = x;
}
if (range2.offset(0, x, 1, 1).getValue() == "Retorn fiança ") {
  var returnOfTheDepositCol = x;
}
if (range2.offset(0, x, 1, 1).getValue() == "Comentari ") {
  var commentsCol = x;
}
  if (range2.offset(0, x, 1, 1).getValue() == "Cancelacions") {
    var cancelCol = x;
  }
    if (range2.offset(0, x, 1, 1).getValue() == "Gestor Avisat") {
      var messageGCol = x;
    }
      if (range2.offset(0, x, 1, 1).getValue() == "Usuari Avisat") {
        var messageUCol = x;
      }
}

ss.getSheets()[0].sort(dateInColumnCol+1); //This code sort descendent by arrival data
column. The +1 is for this function the first column is 1. But the function getRange the first
column is 0

var range = SpreadsheetApp.getActiveSpreadsheet().getSheets()[0].getDataRange();
for (var c = 2; c <= ss.getSheets()[2].getDataRange().getLastColumn(); c++){
  if (ss.getSheets()[2].getRange(1, c).getValue() == "SELECTED") {
    var languageSelected = c;
  }
}
var c = 2;
var report = "";
var nextMonthEntries = "";

```

```

for (var i = range.getRow(); i <
SpreadsheetApp.getActiveSheet().getDataRange().getLastRow(); i++) {
  //var rowValues = new Array(range.getLastColumn());
  var rowValues = new Array();

  for (var x = 0; x < range.getLastColumn(); x++) {
    rowValues[x] = range.getValues()[i][x];
  }
  var rowRange = range.offset(i, 0, 1);
  var origen = range.getValues()[i][origenCol];
  var nameAndSurname = range.getValues()[i][nameAndSurnameCol];
  var email = range.getValues()[i][emailCol];
  var contactData = range.getValues()[i][contactDataCol];
  var dateInColumn = range.getValues()[i][dateInColumnCol];
  var formattedDateInColumn = Utilities.formatDate(dateInColumn, "GMT", "EEE, d MMM
yyyy HH:mm:ss Z");
  var dateOutColumn = range.getValues()[i][dateOutColumnCol];
  var formattedDateOutColumn = Utilities.formatDate(dateOutColumn, "GMT", "EEE, d MMM
yyyy HH:mm:ss Z");
  var numDays = range.getValues()[i][numDaysCol];
  var numPerson = range.getValues()[i][numPersonCol];
  var numAdults = range.getValues()[i][numAdultsCol];
  var numChild = range.getValues()[i][numChildCol];
  var country = range.getValues()[i][countryCol];
  var costToTheCustomer = range.getValues()[i][costToTheCustomerCol];
  var depositPayment = range.getValues()[i][depositPaymentCol];
  var balance = range.getValues()[i][balanceCol];
  var paid = range.getValues()[i][paidCol];
  var cleaningExpenditure = range.getValues()[i][cleaningExpenditureCol];
  var managementExpenditure = range.getValues()[i][managementExpenditureCol];
  var managementTax = range.getValues()[i][managementTaxCol];
  var managementGain = range.getValues()[i][managementGainCol];
  var managementFinances = range.getValues()[i][managementFinancesCol];
  var managementChecklist = range.getValues()[i][managementChecklistCol];
  var managementReview = range.getValues()[i][managementReviewCol];
  var calendarUpdated = range.getValues()[i][calendarUpdatedCol];
  var returnOfTheDeposit = range.getValues()[i][returnOfTheDepositCol];
  var comments = range.getValues()[i][commentsCol];
  var cancel = range.getValues()[i][cancelCol];
  var messageG = range.getValues()[i][messageGCol]; //We can no use state variables to
check this because everytime the script runs we lost the variables value
  var messageU = range.getValues()[i][messageUCol]; //For that reason i need to create a
state in table, and in this way the flat manager is updated too.
  //var today = new Date(new Date().getTime() + 19 * 24 * 60 * 60 * 1000); today fake to
simulate 5 days after change of month
  var today = new Date();
  var emailAddress = "";
  var currentMonth = today.getMonth();
  var nextWeek = new Date(today.getTime() + 7 * 24 * 60 * 60 * 1000);
  var tomorrow = new Date(today.getTime() + 1 * 24 * 60 * 60 * 1000);
  var fiveDaysMore = new Date(today.getTime() + 5 * 24 * 60 * 60 * 1000);

```

```

var lastyear = new Date(today.getTime() + 365 * 24 * 60 * 60 * 1000);
var count = 0;
var hour = today.getHours();
var languageColumn = 2;
var nextMonth = (today.getMonth()+1)%12; //isn't enought adding 1, because Month start
count since 0 to 11
var yearToCompare = today.getYear();
var actualYear = today.getYear();
var yearAndMonthLastRow = "";
var nextMonthEntriesRows = "";

// if (nextMonth == 0){var yearToCompare = today.getYear()+1;} //this if is to prevent the
change of year just in case. This case is only if sheet contains a on year old entrie
if (dateInColumn.getMonth() == nextMonth && dateInColumn.getYear() ==
yearToCompare){
  for (var e = nameAndSurnameCol; e <= numPersonCol; e++){
    nextMonthEntriesRows += "<td style='padding:5px>" + rowValues[e] + "</td>";
  }
  nextMonthEntries += "<tr>" + nextMonthEntriesRows + "</tr>";
}

if (dateOutColumn.valueOf() >= today.valueOf()){ //start set row color

//check overlap data
if (i == 1){
  var dateOutColumn = range.getValues()[i][dateOutColumnCol];
  var ComparationDateIn = range.getValues()[i+1][dateInColumnCol];

  if (ComparationDateIn > dateOutColumn){
    //rowRange.setBackground("green");
  }else{
    //rowRange.setBackground("red");
    for (c = 2; c <= ss.getSheets()[1].getDataRange().getLastColumn(); c++){
      emailAddress += ss.getSheets()[1].getRange(2,c).getValue() + ", ";
    }
    var subject = ss.getSheets()[2].getRange(9,languageSelected).getValue();
    var message = ss.getSheets()[2].getRange(8,languageSelected).getValue() + "\n" +
nameAndSurname + "\n" + formattedDateInColumn + "\n" + formattedDateOutColumn;
    MailApp.sendEmail(emailAddress, subject, message);
  }
}
if (i > 1 && i <= SpreadsheetApp.getActiveSheet().getDataRange().getLastRow()-2){
  var dateInColumn = range.getValues()[i][dateInColumnCol];
  var dateOutColumn = range.getValues()[i][dateOutColumnCol];
  var ComparationDateIn = range.getValues()[i+1][dateInColumnCol];
  var ComparationDateOut = range.getValues()[i-1][dateOutColumnCol];

  if (ComparationDateIn > dateOutColumn && ComparationDateOut < dateInColumn){
    //rowRange.setBackground("green");
    ///range.setBackground("green")[[dateOutColumnCol]];
  }
}
}

```

```

}else{
  //rowRange.setBackgroundColor("red");
  ///range.setBackgrounds("red")[i][dateOutColumnCol];
  for (c = 2; c <= ss.getSheets()[1].getDataRange().getLastColumn(); c++){
    emailAddress += ss.getSheets()[1].getRange(2,c).getValue() + ", ";
  }
  var subject = ss.getSheets()[2].getRange(9,languageSelected).getValue();
  var message = ss.getSheets()[2].getRange(8,languageSelected).getValue() + "\n" +
nameAndSurname + "\n" + formattedDateInColumn + "\n" + formattedDateOutColumn;
  MailApp.sendEmail(emailAddress, subject, message);
}
}

if (i == SpreadsheetApp.getActiveSheet().getDataRange().getLastRow()-1){
  var dateInColumn = range.getValues()[i][dateInColumnCol];
  var ComparationDateOut = range.getValues()[i-1][dateOutColumnCol];

  if (ComparationDateOut < dateInColumn){
    //rowRange.setBackgroundColor("green");
  }else{
    //rowRange.setBackgroundColor("red");
    for (c = 2; c <= ss.getSheets()[1].getDataRange().getLastColumn(); c++){
      emailAddress += ss.getSheets()[1].getRange(2,c).getValue() + ", ";
    }
    var subject = ss.getSheets()[2].getRange(9,languageSelected).getValue();
    var message = ss.getSheets()[2].getRange(8,languageSelected).getValue() + "\n" +
nameAndSurname + "\n" + formattedDateInColumn + "\n" + formattedDateOutColumn;
    MailApp.sendEmail(emailAddress, subject, message);
  }
}

//end of check overlap data

if (cancel == false){

  //if calendar exist use the ID of existent calendar in the cell else create it and write in cell
  var calendarID =
SpreadsheetApp.getActiveSpreadsheet().getSheets()[1].getSheetValues(1, 2, 1, 1).valueOf();
  if (calendarID == ""){
    var calendars = CalendarApp.createCalendar(ss.getName(), {summary:
ss.getSheets()[2].getRange(19,languageSelected).getValue(), color:
CalendarApp.Color.BLUE});
    var calendarID = calendars.getId();
  }
  SpreadsheetApp.getActiveSpreadsheet().getSheets()[1].getRange("B1").setValue(calendarID)
;
  }else{

    var cal = CalendarApp.getCalendarById(calendarID);
    var events = cal.getEvents(dateInColumn, dateOutColumn, {search:
nameAndSurname});
}
}

```

```

if (events.length == 0){
    var event = cal.getEventSeriesById(calendarID);
    var title = nameAndSurname;
    var desc = ss.getSheets()[2].getRange(16,languageSelected).getValue() +
nameAndSurname;
    var loc = ss.getSheets()[1].getRange(4,2).getValue();
    var events = cal.createEvent(desc, dateInColumn, dateOutColumn, {location: loc,
description: desc});
}
}

if (range.getValues()[i][dateInColumnCol].length != 0 &&
range.getValues()[i][dateOutColumnCol].length != 0 && range.getValues()[i][origenCol].length
!= 0 && range.getValues()[i][contactDataCol].length != 0) {
    if (calendarUpdated == true ) {
        if (cancel == true) {
            rowRange.setBackground("gray");
            rowRange.setFontLine("line-through");
            range.getCell(i+1, commentsCol+1).setValue("pinto de gris tachado por estar
cancelado y con el calendario actualizado");
        }else if (cancel == false){
            rowRange.setBackground("white");
            range.getCell(i+1, commentsCol+1).setValue("pinto de blanco pq tiene dateInColumn
dateOutColumn y origen con el calendario actualizado además se ha comprobado si estaba
dentro de la semana para entrar y no se ha dado alguno de ese caso");
        }
        if (nextWeek >= dateInColumn && tomorrow < dateOutColumn) {
            if (balance > 0){
                rowRange.setBackground("yellow");
                range.getCell(i+1, commentsCol+1).setValue("pinto de amarillo pq falta por pagar
estando en la semana antes (o durante) de la llegada sin estar cancelado y con el calendario
actualizado");
                if (messageG != today){
                    for (c = 2; c <= ss.getSheets()[1].getDataRange().getLastColumn(); c++){
                        emailAddress += ss.getSheets()[1].getRange(2,c).getValue() + ", ";
                    }
                    var daysDifference = dateOutColumn.getDate() - today.getDate();
                    var subject = balance + " " +
ss.getSheets()[2].getRange(10,languageSelected).getValue() + daysDifference;
                    var message = ss.getSheets()[2].getRange(7,languageSelected).getValue() + "\n"
+ nameAndSurname + "\n" + formattedDateInColumn + "\n" + formattedDateOutColumn + "\n"
+ contactData + "\n" + country + "\n" + email;
                    MailApp.sendEmail(emailAddress, subject, message);
                    range.getCell(i+1, 27+1).setValue(today);
                }
            }
        }else if (balance < 0){
//something wrong?
            rowRange.setBackground("yellow");
            range.getCell(i+1, commentsCol+1).setValue("pinto de amarillo pq se ha pagado de
más estando en la semana antes (o durante) de la llegada sin estar cancelado y con el
calendario actualizado");
        }else if (balance == 0){
        }
    }
}

```

```

rowRange.setBackgroundColor("white");
range.getCell(i+1, commentsCol+1).setValue("pinto de blanco pq se ha pagado la estancia estando en la semana antes (o durante) de la llegada sin estar cancelado y con el calendario actualizado");
if (today >= dateInColumn && today <= dateOutColumn) {
    rowRange.setFontColor("green");
    range.getCell(i+1, commentsCol+1).setValue("pinto de verde las letras pq se ha pagado la estancia durante la semana en la que está ocupado el piso sin estar cancelado y con el calendario actualizado");
}
}
}else if (tomorrow == dateOutColumn){
    if (balance == 0){
        rowRange.setBackgroundColor("yellow");
        range.getCell(i+1, commentsCol+1).setValue("pinto de amarillo pq mañana se van del piso sin estar cancelado y con el calendario actualizado si se ha pagado y además se manda un mail al operario de limpieza para pedirle que esté allí al día siguiente");
        for (var c = 2; c <=
SpreadsheetApp.getActiveSpreadsheet().getSheets()[1].getDataRange().getLastColumn();
c++){
            emailAddress +=
SpreadsheetApp.getActiveSpreadsheet().getSheets()[1].getRange(3,c).getValue() + ", ";
        } //end of constructing email list address to send cleaning service mail
        var subject =
SpreadsheetApp.getActiveSpreadsheet().getSheets()[2].getRange(27,languageSelected).getValue();
        var message = ss.getSheets()[2].getRange(28,languageSelected).getValue() + "\n" +
dateOutColumn;
        MailApp.sendEmail(emailAddress, subject, message);
        for (var c = 2; c <=
SpreadsheetApp.getActiveSpreadsheet().getSheets()[1].getDataRange().getLastColumn();
c++){
            emailAddress +=
SpreadsheetApp.getActiveSpreadsheet().getSheets()[1].getRange(2,c).getValue() + ", ";
        } //end of constructing email list address to send advertice mail
        to manager
        var subject =
ss.getSheets()[2].getRange(14,languageSelected).getValue() + nameAndSurname;
        var message =
ss.getSheets()[2].getRange(15,languageSelected).getValue() + nameAndSurname;
        MailApp.sendEmail(emailAddress, subject, message);

}else if (balance != 0){
    rowRange.setBackgroundColor("red");
    range.getCell(i+1, commentsCol+1).setValue("pinto de rojo pq mañana se van del piso sin estar cancelado y con el calendario actualizado sin haber pagado!! Se manda un mail al gestor");
    for (var c = 2; c <=
SpreadsheetApp.getActiveSpreadsheet().getSheets()[1].getDataRange().getLastColumn();
c++){
        emailAddress +=

```

```

SpreadsheetApp.getActiveSpreadsheet().getSheets()[1].getRange(2,c).getValue() + ", ";
} //end of constructing email list address to send advertice mail to manager
var daysDifference = dateOutColumn.getDate() - today.getDate();
var subject = ss.getSheets()[2].getRange(29,languageSelected).getValue();
var message = ss.getSheets()[2].getRange(30,languageSelected).getValue() + "\n"
+ nameAndSurname + "\n" + formattedDateInColumn + "\n" + formattedDateOutColumn + "\n"
+ contactData + "\n" + country + "\n" + email + "\n" + daysDifference + "\n" + balance;
MailApp.sendEmail(emailAddress, subject, message);
}
}else if (today == dateOutColumn){
if (managementChecklist == false) {
rowRange.setBackgroundColor("red");
range.getCell(i+1, commentsCol+1).setValue("pinto de rojo pq hoy es
dateOutColumn y o han pagado ya, o me quedo la fianza, así que ya da igual y he supuesto
que el limipador me ha respondido positivamente al appointment pero el checklist está a
false");
}else if (managementChecklist == true){
if (managementReview == true && returnOfTheDeposit == true) {
rowRange.setBackgroundColor("white");
range.getCell(i+1, commentsCol+1).setValue("pinto de blanco pq hoy es
dateOutColumn y o han pagado ya, o me quedo la fianza, así que ya da igual y he supuesto
que el limipador me ha respondido positivamente al appointment, el checklist está a true y
además managementReview == true && returnOfTheDeposit == true");
} else if (managementReview != true || returnOfTheDeposit != true) {
rowRange.setBackgroundColor("orange");
range.getCell(i+1, commentsCol+1).setValue("mando mail al gestor para que
haga algo para cambiar el retorno de fianza y la reseña ya que el checklist está OK y además
pinto de amarillo pq hoy es dateOutColumn y o han pagado ya, o me quedo la fianza, así que
ya da igual y he supuesto que el limipador me ha respondido positivamente al appointment,
además el checklist está a true PERO managementReview != true || returnOfTheDeposit != true");
for (var c = 2; c <=
SpreadsheetApp.getActiveSpreadsheet().getSheets()[1].getDataRange().getLastColumn();
c++){
emailAddress +=
SpreadsheetApp.getActiveSpreadsheet().getSheets()[1].getRange(2,c).getValue() + ", ";
} //end of constructing email list address to send advertice mail to manager
var subject = nameAndSurname + " " +
ss.getSheets()[2].getRange(17,languageSelected).getValue();
var message = ss.getSheets()[2].getRange(18,languageSelected).getValue() +
managementChecklist;
MailApp.sendEmail(emailAddress, subject, message);
}
}
}
}
}else if (calendarUpdated == false){
if (cancel == true) {
rowRange.setBackgroundColor("yellow");
rowRange.setFontLine("line-through");
range.getCell(i+1, commentsCol+1).setValue("pinto de amarillo tachado por estar

```

```

cancelado y con el calendario NO actualizado");
}else if (cancel == false){
    rowRange.setBackgroundColor("yellow");
    range.getCell(i+1, commentsCol+1).setValue("pinto de amarillo pq tiene
dateInColumn dateOutColumn y origen pero con el calendario NO actualizado y NO está
cancelado");
}
}
}else if (range.getValues()[i][dateInColumnCol].length == 0 &&
range.getValues()[i][dateOutColumnCol].length == 0 && range.getValues()[i][origenCol].length
== 0 && range.getValues()[i][contactDataCol].length == 0) {
    rowRange.setBackgroundColor("yellow");
    range.getCell(i+1, commentsCol+1).setValue("pinto de amarillo pq les falta los datos de
contacto, la fecha de entrada, el origen o la fecha de salida");
}

if (today == dateInColumn && messageG != today){
    for (var c = 2; c <=
SpreadsheetApp.getActiveSpreadsheet().getSheets()[1].getDataRange().getLastColumn());
c++){
        emailAddress +=
SpreadsheetApp.getActiveSpreadsheet().getSheets()[1].getRange(2,c).getValue() + ", ";
        } //end of constructing email list address to send advertice mail to manager
        var subject = ss.getSheets()[2].getRange(3,languageSelected).getValue() +
nameAndSurname;
        var message = ss.getSheets()[2].getRange(4,languageSelected).getValue() +
nameAndSurname;
        MailApp.sendEmail(emailAddress, subject, message);

        range.getCell(i+1, 27+1).setValue(today);
    }
}
if ((dateOutColumn.valueOf() < today.valueOf() && managementChecklist == true &&
balance <= 0) || cancel == true) {//If date is past, and is checklist is OK and is paid, OR IS
CANCEL, move to another page. In order to be accurate .valueOf give us the number of
milliseconds of the date variable.
    // The code below will copy from active sheet the first row to another sheet
    var sheet = SpreadsheetApp.getActiveSpreadsheet();

    sheet.getSheets()[0].getRange(i+1, 1, 1,
SpreadsheetApp.getActiveSpreadsheet().getSheets()[0].getDataRange().getLastColumn()).co
pyTo(sheet.getSheets()[3].getRange(sheet.getSheets()[3].getLastRow()+1, 1, 1,
SpreadsheetApp.getActiveSpreadsheet().getSheets()[0].getDataRange().getLastColumn()),
{contentsOnly: false});

    // The code below will erase the copied row and positioned the next row in his spot
    sheet.getSheets()[0].deleteRow(2);

    //we need to set one step before the i variable to point to new set of values client in the
new row
    i--;
}

```

```

} //end of else
} //end of for to record all cells

report = "<head><style>table, th, td {border: 1px solid black; border-collapse: collapse;}th, td
{padding: 15px;}</style></head><table><tr><th>Nom i cognom</th><th>Email client</th><th>Dades de contacte (telfs, adreces,...)</th><th>Data d'entrada</th><th>Data de sortida</th><th># dies</th><th># persones</th></tr>" +
nextMonthEntries + "</table>";

if (fiveDaysMore.getMonth() == nextMonth && fiveDaysMore.getDate() == 1){
  for (c = 2; c <= ss.getSheets()[1].getDataRange().getLastColumn(); c++){
    emailAddress += ss.getSheets()[1].getRange(3,c).getValue() + ", ";
  } //end of constructing email list address to send calendar
  var subject = ss.getSheets()[2].getRange(20,languageSelected).getValue();
  MailApp.sendEmail(emailAddress, subject, report, {htmlBody: report});
} //end of if to determinate if is the date to send mail to cleaner with next month row entries
for (var c = 0; c <
SpreadsheetApp.getActiveSpreadsheet().getSheets()[0].getDataRange().getLastColumn();
c++) {
  SpreadsheetApp.getActiveSpreadsheet().getSheets()[3].setColumnWidth(c+1,
SpreadsheetApp.getActiveSpreadsheet().getSheets()[0].getColumnWidth(c+1));
}
} //end of function

```

Al principi cal comprovar que la fulla "Past_bookings" existeixi i de no ser així crear-la copiant la primera fila i congelant aquesta i les 3 primeres columnes. Donant-li així el mateix aspecte que la fulla principal del pis.

Fem que la fulla principal sigui l'activa per si en algun moment ens cal referir-nos a ella com "Fulla activa" amb la funció "getActiveSheet()". I afitem el rang de cel·les fins l'última fila i columna. Amb range2 mapejarem el número de columna amb el nom de la columna evitant hardcodejar el número de columna i així no tenir problemes per si en algun moment calgués treure o afegir columnes a la fulla.

Ordenem per data d'entrada. I entrem al for que recorre totes les files. Dins d'aquest, hi trobem el for que recorre totes les columnes dins d'una fila i les guarda a l'array rowValues. El valor de cada cel·la és assignat a una variable amb un nom prou descriptiu per poder-les rescatar i utilitzar al codi amb facilitat.

Veiem un if el qual detecta si la data d'entrada y de sortida de la reserva que està mirant en aquest moment l'script està compresa en el més i any actuals. El sentit de que la variable nextMonthEntriesRows estigui definida dins del for que recorre la taula i la variable nextMonthEntries estigui definida fora és que la primera variable construeix una taula d'una fila i 9 columnes que, un cop ha volcat la informació a la segona variable, necessitem resetear i tornar-la a possar sense cap valor. El motiu de que la segona variable estigui definida fora és que va constraint per cada fila que toca una taula en format HTML i va guardant la informació acumulant-la fins al final de la taula que és quan es crea la resta del cos HTML i s'envia el mail

al responsable de neteja si s'escau. Ja per què hi hagi alguna modificació dins del més actual o potser per què falten 5 dies pel més que ve.

Seguidament comprova la col·lisió entre dates. Si en detecta alguna, envia un mail al gestor per que ho arregli.

L'aplicació comprova si ja existeix un calendari pel pis i si no existeix, el crea i l'apunta a la fulla de "Settings". Acte seguit crea una cita en aquest calendari amb la informació de la fila en la que es troba.

Comencem amb les condicions per pintar les files que calguin:

1. Si hi ha data d'entrada, de sortida però no hi ha dades del contacte i els calendaris no estan actualitzats -> Pinta de groc
2. Si hi ha data d'entrada, de sortida, hi ha dades del contacte però els calendaris no estan actualitzats i està cancelat -> Pinta de groc i tatxa les lletres
3. Si hi ha data d'entrada, de sortida, hi ha dades del contacte, els calendaris estan actualitzats i està cancelat -> Pinta de gris i tatxa les lletres
4. Si hi ha data d'entrada, de sortida, hi ha dades del contacte, els calendaris estan actualitzats, encara no està tot pagat i la data d'avui està dins del període d'una setmana abans de l'arribada al pis -> Pinta de groc i si encara no ha enviat un missatge d'avís d'impagament al gestor, li envia i canvia la columna de gestor avisat amb la data del dia que ha enviat l'avís per evitar el bombardeig d'emails.
5. Si hi ha data d'entrada, de sortida, hi ha dades del contacte, els calendaris estan actualitzats, està tot pagat i la data d'avui està dins del període de l'estada al pis -> Pinta de verd les lletres i de blanc les cel·les i si encara no has enviat un missatge d'avís d'arribada al gestor, envia-li i canvia la columna de gestor avisat amb la data del dia que ha enviat l'avís . Si demà és el dia que marxa, també se li envia un mail al gestor i es pinta la fila de color groc o vermell si el pagament està bé o hi ha quelcom malament. Si el pagament és correcte se li envia un mail al gestor i un altre al personal de neteja en espera de confirmació. Si el pagament no és correcte se li envia un mail només al gestor avisant-lo de la situació.
6. Si avui coincideix amb la data de sortida, es mira el checkList. Si el checkList està a false es pinta la fila de vermell donat que significa que al pis hi ha alguna cosa malament i s'ha de reparar. Si el checkList està a true, i el retorn de la fiança i la resenya també, es pinta de blanc, si alguna de les dues no estigués ok, es pinta de taronja i s'envia un mail al gestor.

I sempre s'envia un mail al gestor el dia que arriba un nou client al pis.

El següent if ens separa les entrades que ja han sigut completades per data de sortida o bé les cancel·lades i les envia a la pàgina "Past_bookings". El procés copia la cel·la a l'altre fulla i elimina la fila fent pujar les següents 1 posició.

L'últim bloc d'aquest apartat crea el report per l'operari de neteja generant una taula en HTML, ja que era la única forma de poder crear una taula de forma automàtica a un mail de gmail. Si coincideix en que falten 5 dies per l'entrada del nou més, s'enviarà i si no, no.

```
function onOpen() {
  var ui = SpreadsheetApp.getUi();
  ui.createMenu("Crear/Editar/Cancelar/Resum/Llenguatge/Executar script")
    .addSubMenu(ui.createMenu("Nous")
      .addItem('Nova entrada', 'menulItem1')
      .addSeparator()
      .addSubMenu(ui.createMenu("Existents")
        .addItem('Editar', 'menulItem2')
        .addSeparator()
        .addItem('Resum', 'menulItem3')
        .addSeparator()
        .addSubMenu(ui.createMenu("Llenguatge")
          .addItem('English', 'englishMenulItem1')
          .addSeparator()
          .addItem('Català', 'catalaMenulItem1')
          .addSeparator()
          .addItem('Español', 'espanolMenulItem1'))
        .addSeparator()
        .addItem('Executar script', 'setRowColors')
      ).addToUi());
}
```

Al obrir l'aplicació es creen els punts de menú per introduïr una nova reserva, editar-ne alguna d'existents mitjançant el número identificador del client, generar el resum econòmic, escollir l'idioma en que s'envien els missatges o es crean els resums i fins i tot executar l'script principal.

```
function englishMenulItem1() {
  var ss = SpreadsheetApp.getActiveSpreadsheet();
  ss.getSheets()[2].getRange(1, 2).setValue("SELECTED");
  ss.getSheets()[2].getRange(1, 3).setValue("");
  ss.getSheets()[2].getRange(1, 4).setValue("");
}
```

```
function catalaMenulItem1() {
  var ss = SpreadsheetApp.getActiveSpreadsheet();
  ss.getSheets()[2].getRange(1, 2).setValue("");
  ss.getSheets()[2].getRange(1, 3).setValue("");
  ss.getSheets()[2].getRange(1, 4).setValue("SELECTED");
}
```

```
function espanolMenulItem1() {
  var ss = SpreadsheetApp.getActiveSpreadsheet();
  ss.getSheets()[2].getRange(1, 2).setValue("");
  ss.getSheets()[2].getRange(1, 3).setValue("SELECTED");
  ss.getSheets()[2].getRange(1, 4).setValue("");
```

{}

Segons l'opció que s'esculli s'escriu sobre la columna de cada idioma de la fulla "Snippets" la paraula "SELECTED" i s'esborra el que hi hagi a altres columnes d'aquella fila. De manera que el número de columna queda "marcat" per a que es pugui utilitzar per fer referència a la frase en el llenguatge que es vol.

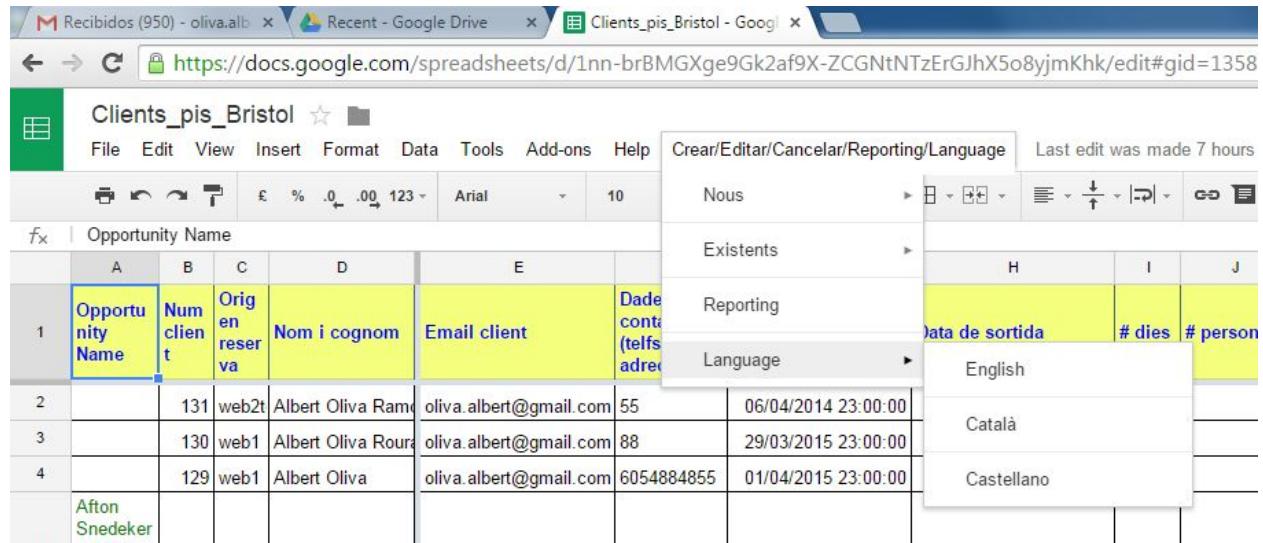


Fig.16 Selecció del llenguatge.

```

function menuitem1() {
//I decided to implement a UI application , so we do not create additional documents (Form )
in the drive of the user and the app is lighter
  var app = UiApp.createApplication();
  var ss = SpreadsheetApp.getActiveSpreadsheet();
  ss.setActiveSheet(ss.getSheets()[0]);
  for (var c = 2; c <= ss.getSheets()[2].getDataRange().getLastColumn(); c++){
    if (ss.getSheets()[2].getRange(1, c).getValue() == "SELECTED") {
      var languageSelected = c;
    }
  }
  app.setTitle(ss.getSheets()[2].getRange(21,languageSelected).getValue());
  var today = new Date();
  var tomorrow = new Date(today.getTime() + 1 * 24 * 60 * 60 * 1000);
  var date1Mod = new Date(today.setHours(12, 00, 00)); //today fake to simulate 5 days after
change of month
  var date2Mod = new Date(tomorrow.setHours(10, 00, 00)); //today fake to simulate 5 days
after change of month

  //in order to calculate the next num entrie i create an array with all the numbers and extract
the max with Math function
  var range = SpreadsheetApp.getActiveSheet().getDataRange();
  var lastColumn = SpreadsheetApp.getActiveSheet().getLastColumn();
}

```

```

var range2 = SpreadsheetApp.getActiveSheet().getRange(1,1,1,lastColumn);
for (var x = 0; x < range2.getLastColumn(); x++) {
  if (range2.offset(0, x, 1, 1).getValue() == "Num client ") {
    var numClient = x;
  }
}
var array = new Array();
for (var x = 1; x < range.getLastRow(); x++) {
  array[x-1] = range.getValues()[x][numClient];
}

var panel = app.createVerticalPanel();

var max = Math.max.apply(Math, array); // number for new registre

var labelOportunity = app.createLabel("Oportunity name");
var textOportunity = app.createTextBox();
var labelnum = app.createLabel("Número de client");
var num = app.createTextBox().setText(max+1);
var label1 = app.createLabel("Origen reserva");
var text1 = app.createTextBox();
var label2 = app.createLabel("Nom i cognom ");
var text2 = app.createTextBox();
var label3 = app.createLabel("Email client ");
var text3 = app.createTextBox();
var label4 = app.createLabel("Dades de contacte (telfs, adreces,...) ");
var text4 = app.createTextBox();
var labeldate1 = app.createLabel("Data d'entrada");
// var date1 = app.createDateBox().setValue(date1Mod);
var dFormat = UiApp.DateTimeFormat.DATE_TIME_MEDIUM;
var date1 = app.createDateBox().setFormat(dFormat).setValue(date1Mod);
var labeldate2 = app.createLabel("Data de sortida");
var date2 = app.createDateBox().setFormat(dFormat).setValue(date2Mod);
// var label5 = app.createLabel("# dies");
// var text5 = app.createTextBox();
// var label6 = app.createLabel("# persones");
// var text6 = app.createTextBox();
var label7 = app.createLabel("# adults");
var text7 = app.createTextBox();
var label8 = app.createLabel("# nens");
var text8 = app.createTextBox();
var label9 = app.createLabel("Pais");
var text9 = app.createTextBox();
var label10 = app.createLabel("Cost pel client");
var text10 = app.createTextBox();
var label11 = app.createLabel("Paga i senyal");
var text11 = app.createTextBox();
// var label12 = app.createLabel("Saldo");
// var text12 = app.createTextBox();
var label13 = app.createLabel("Pagat");
var text13 = app.createTextBox();

```

```
var label14 = app.createLabel("Despeses de neteja");
var text14 = app.createTextBox();
var label15 = app.createLabel("Despeses de gestió");
var text15 = app.createTextBox();
var label16 = app.createLabel("Impostos");
var text16 = app.createTextBox();
// var label17 = app.createLabel("Guanys");
// var text17 = app.createTextBox();
var label18 = app.createLabel("Fiança");
var text18 = app.createTextBox();
var text19 = app.createCheckBox("Checklist");
var text20 = app.createCheckBox("Ressenya");
var text21 = app.createCheckBox("Calendaris actualitzats");
var text22 = app.createCheckBox("Retorn fiança");
var label23 = app.createLabel("Comentaris");
var text23 = app.createTextBox();
var text24 = app.createCheckBox("Cancelacions");

textOportunity.setName("TextOportunity");
num.setName("NumClient");
text1.setName("TextBox1");
text2.setName("TextBox2");
text3.setName("TextBox3");
text4.setName("TextBox4");
date1.setName("Date1");
date2.setName("Date2");
// text5.setName("TextBox5");
// text6.setName("TextBox6");
text7.setName("TextBox7");
text8.setName("TextBox8");
text9.setName("TextBox9");
text10.setName("TextBox10");
text11.setName("TextBox11");
// text12.setName("TextBox12");
text13.setName("TextBox13");
text14.setName("TextBox14");
text15.setName("TextBox15");
text16.setName("TextBox16");
// text17.setName("TextBox17");
text18.setName("TextBox18");
text19.setName("TextBox19");
text20.setName("TextBox20");
text21.setName("TextBox21");
text22.setName("TextBox22");
text23.setName("TextBox23");
text24.setName("TextBox24");

var button = app.createButton("Ok");
panel.setSpacing(5);
```

```
panel.add(labelOportunity);
panel.add(textOportunity);
panel.add(labelnum);
panel.add(num);
panel.add(label1);
panel.add(text1);
panel.add(label2);
panel.add(text2);
panel.add(label3);
panel.add(text3);
panel.add(label4);
panel.add(text4);
panel.add(labeldate1);
panel.add(date1);
panel.add(labeldate2);
panel.add(date2);
// panel.add(label5);
// panel.add(text5);
// panel.add(label6);
// panel.add(text6);
panel.add(label7);
panel.add(text7);
panel.add(label8);
panel.add(text8);
panel.add(label9);
panel.add(text9);
panel.add(label10);
panel.add(text10);
panel.add(label11);
panel.add(text11);
// panel.add(label12);
// panel.add(text12);
panel.add(label13);
panel.add(text13);
panel.add(label14);
panel.add(text14);
panel.add(label15);
panel.add(text15);
panel.add(label16);
panel.add(text16);
// panel.add(label17);
// panel.add(text17);
panel.add(label18);
panel.add(text18);
panel.add(text19);
panel.add(text20);
panel.add(text21);
panel.add(text22);
panel.add(label23);
panel.add(text23);
panel.add(text24);
```

```

panel.add(button);

var handler = app.createServerHandler("submitButton");
button.addClickHandler(handler);
handler.addCallbackElement(panel);

var scroll = app.createScrollPanel().setPixelSize(500, 300);
scroll.add(panel);
app.add(scroll);

var doc = SpreadsheetApp.getActiveSpreadsheet();
doc.show(app);

}

```

Es va decidir crear un formulari en forma de finestra emergent per no crear més documents. D'aquesta manera l'aplicació és més compacta i genera menys brossa.

```

function submitButton(e) {

  for (var c = 2; c <=
SpreadsheetApp.getActiveSpreadsheet().getSheets()[2].getDataRange().getLastColumn();
c++){
    if (SpreadsheetApp.getActiveSpreadsheet().getSheets()[2].getRange(1, c).getValue() ==
"SELECTED") {
      var languageSelected = c;
    }
  }

  var days = new Date(e.parameter.Date2 - e.parameter.Date1);
  var dateTemp1 = e.parameter.Date1;
  var date1 = new Date(dateTemp1.setHours(13, 00, 00));
  var dateTemp2 = e.parameter.Date2;
  var date2 = new Date(dateTemp2.setHours(11, 00, 00));
  var people = +e.parameter.TextBox7 + +e.parameter.TextBox8;
  var autoBalance = +e.parameter.TextBox10 - +e.parameter.TextBox11 -
+e.parameter.TextBox13;
  var autoManagementGain = +e.parameter.TextBox10 - +e.parameter.TextBox14 -
+e.parameter.TextBox15 - +e.parameter.TextBox16;
  var app = UiApp.getActiveApplication();

  var textOportunity = e.parameter.TextOportunity;
  var numValue = e.parameter.NumClient;
  var textValue = e.parameter.TextBox1;
  var textValue2 = e.parameter.TextBox2;
  var textValue3 = e.parameter.TextBox3;
  var textValue4 = e.parameter.TextBox4;
// var dateValue = e.parameter.Date1;

```

```

var dateValue = date1;
// var dateValue2 = e.parameter.Date2;
var dateValue2 = date2;
var textValue5 = days.getDate();
var textValue6 = people;
var textValue7 = e.parameter.TextBox7;
var textValue8 = e.parameter.TextBox8;
var textValue9 = e.parameter.TextBox9;
var textValue10 = e.parameter.TextBox10;
var textValue11 = e.parameter.TextBox11;
var textValue12 = autoBalance;
var textValue13 = e.parameter.TextBox13;
var textValue14 = e.parameter.TextBox14;
var textValue15 = e.parameter.TextBox15;
var textValue16 = e.parameter.TextBox16;
var textValue17 = autoManagementGain;
var textValue18 = e.parameter.TextBox18;
var textValue19 = e.parameter.TextBox19;
var textValue20 = e.parameter.TextBox20;
var textValue21 = e.parameter.TextBox21;
var textValue22 = e.parameter.TextBox22;
var textValue23 = e.parameter.TextBox23;
var textValue24 = e.parameter.TextBox24;
var textValue25 = false;
var textValue26 = false;

var sheet = SpreadsheetApp.getActiveSpreadsheet();
var lastCol = sheet.getSheets()[0].getLastColumn();
var lastRow = sheet.getSheets()[0].getLastRow();

var source = sheet.getSheets()[0]
var destination = sheet.getSheets()[0]
var range = source.getRange(lastRow, 1, 1, lastCol);

// This copies the formatting in last column and last row added
var newArray = new Array();
newArray = [textOpportunity, numValue, textValue, textValue2, textValue3, textValue4,
dateValue, dateValue2, textValue5, textValue6, textValue7, textValue8, textValue9,
textValue10, textValue11, textValue12, textValue13, textValue14, textValue15, textValue16,
textValue17, textValue18, textValue19, textValue20, textValue21, textValue22, textValue23,
textValue24, textValue25, textValue26];
sheet.appendRow(newArray);
range.copyFormatToRange(destination, 1, lastCol, lastRow+1, lastRow+1);
sheet.getSheets()[0].sort(7);//This code sort descendent by column 7
sheet.getSheets()[0].getRange(lastRow+1, 1, 1, lastCol).setBorder(true, true, true, true, true);

var today = new Date();

Browser.msgBox(SpreadsheetApp.getActiveSpreadsheet().getSheets()[2].getRange(31,languageSelected).getValue());

```

```

for (var x = 0; x <
SpreadsheetApp.getActiveSheet().getRange(1,1,1,SpreadsheetApp.getActiveSheet().getLast
Column()).getLastColumn(); x++) {
    if
(SpreadsheetApp.getActiveSheet().getRange(1,1,1,SpreadsheetApp.getActiveSheet().getLast
Column()).offset(0, x, 1, 1).getValue() == "Nom i cognom ") {
        var nameAndSurnameCol = x;
    }
    if
(SpreadsheetApp.getActiveSheet().getRange(1,1,1,SpreadsheetApp.getActiveSheet().getLast
Column()).offset(0, x, 1, 1).getValue() == "Data d'entrada ") {
        var dateInColumnCol = x;
    }
    if
(SpreadsheetApp.getActiveSheet().getRange(1,1,1,SpreadsheetApp.getActiveSheet().getLast
Column()).offset(0, x, 1, 1).getValue() == "# persones ") {
        var numPersonCol = x;
    }
    if
(SpreadsheetApp.getActiveSheet().getRange(1,1,1,SpreadsheetApp.getActiveSheet().getLast
Column()).offset(0, x, 1, 1).getValue() == "Cancelacions") {
        var cancelCol = x;
    }
}

if (autoBalance == 0){
    var emailAddress = "";
    for (var c = 2; c <=
SpreadsheetApp.getActiveSpreadsheet().getSheets()[1].getDataRange().getLastColumn());
c++){
        emailAddress +=
SpreadsheetApp.getActiveSpreadsheet().getSheets()[1].getRange(2,c).getValue() + ", ";
    } //end of constructing email list address to send notification that flat is paid
    var subject =
SpreadsheetApp.getActiveSpreadsheet().getSheets()[2].getRange(25,languageSelected).get
Value() + textView2;
    var message =
SpreadsheetApp.getActiveSpreadsheet().getSheets()[2].getRange(26,languageSelected).get
Value() + textView2;
    MailApp.sendEmail(emailAddress, subject, message);
}

if (date1.getMonth() == today.getMonth() && date1.getYear() == today.getYear()){
    var report = "";
    var nextMonth = (today.getMonth()+1)%12; //isn't enought adding 1, because Month start
count since 0 to 11
    var modifiedEntryActualMonthRow = "";
    var modifiedEntryActualMonth = "";
    var actualMonthEntries = "";
    var emailAddress = "";
}

```

```

var i = 1;
var imin = 0;
var imax = 0;

do{
    if
(SpreadsheetApp.getActiveSpreadsheet().getSheets()[0].getDataRange().getValues()[i][dateInColumnCol].getMonth() < today.getMonth() &&
SpreadsheetApp.getActiveSpreadsheet().getSheets()[0].getDataRange().getValues()[i][dateInColumnCol].getYear() == today.getYear())){
    var imin = i;
}
if
(SpreadsheetApp.getActiveSpreadsheet().getSheets()[0].getDataRange().getValues()[i][dateInColumnCol].getMonth() == today.getMonth() &&
SpreadsheetApp.getActiveSpreadsheet().getSheets()[0].getDataRange().getValues()[i][dateInColumnCol].getYear() == today.getYear())){
    var imax = i;
}
i++;
}while
(SpreadsheetApp.getActiveSpreadsheet().getSheets()[0].getDataRange().getValues()[i][dateInColumnCol].getMonth() < nextMonth)
var imin = imin+1;

for (var i = imin; i <= imax; i++) {
    var actualMonthEntriesRows = "";
    for (var e = nameAndSurnameCol; e <= numPersonCol; e++){
        actualMonthEntriesRows += "<td style='padding:5px>" +
SpreadsheetApp.getActiveSpreadsheet().getSheets()[0].getDataRange().getValues()[i][e] +
"</td>";
    }
    actualMonthEntries += "<tr>" + actualMonthEntriesRows + "</tr>";
}

for (var e = nameAndSurnameCol; e <= numPersonCol; e++){
    modifiedEntryActualMonthRow += "<td style='padding:5px>" + newArray[e] + "</td>";
}
modifiedEntryActualMonth += "<tr>" + modifiedEntryActualMonthRow + "</tr>";

report = "<head><style>table, th, td {border: 1px solid black; border-collapse: collapse;}th, td
{padding: 15px;}</style></head><table><tr><th>Nom i cognom</th><th>Email client</th><th>Dades de contacte (telfs, adreces,...)</th><th>Data d'entrada</th><th>Data de sortida</th><th># dies</th><th># persones</th></tr>" +
actualMonthEntries + "</table><h2>" +
SpreadsheetApp.getActiveSpreadsheet().getSheets()[2].getRange(24,languageSelected).getValue() + "</h2><head><style>table, th, td {border: 1px solid black; border-collapse: collapse;}th, td
{padding: 15px;}</style></head><table><tr><th>Nom i cognom</th><th>Email client</th><th>Dades de contacte (telfs, adreces,...)</th><th>Data d'entrada</th><th>Data de sortida</th><th># dies</th><th># persones</th></tr>" + modifiedEntryActualMonth + "</table></body></html>";

```

```

for (var c = 2; c <=
SpreadsheetApp.getActiveSpreadsheet().getSheets()[1].getDataRange().getLastColumn();
c++){
  emailAddress +=
SpreadsheetApp.getActiveSpreadsheet().getSheets()[1].getRange(3,c).getValue() + ", ";
} //end of constructing email list address to send calendar
var subject =
SpreadsheetApp.getActiveSpreadsheet().getSheets()[2].getRange(20,languageSelected).get
Value();
MailApp.sendEmail(emailAddress, subject, report, {htmlBody: report});

} //end of if to create monthly actual report
Browser.msgBox("Ok!");
return app.close();
}

```

Aquest formulari ens permet introduir totes les dades de totes les columnes, excepte algunes que ja calcula per si mateix (com el número de dies segons les dates introduïdes i el número de persones fent la suma d'adults i nens). També permet afegir alguns valors per defecte com el FALSE de la reserva cancel·lada, ja que si es nova el més probable és que no estigui cancel·lada.

S'escriuen les dades al final del full, es genera el marc de la taula, es copia el format de les dades de la fila anterior i s'ordena per data d'entrada.

L'script revisa si la nova reserva està totalment pagada, de ser així, envia un email al gestor per informar-lo. També revisa si la reserva és dintre del més actual i de ser així envia un email al personal de neteja amb el calendari del més actual actualitzat i amb la nova fila que ha editat aquest calendari per que no hi hagi confusions.

```

function menuItem2() {
  // The code below will set the value of name to the name input by the user, or 'cancel'

  var app2 = UiApp.createApplication();
  var ss = SpreadsheetApp.getActiveSpreadsheet();
  ss.setActiveSheet(ss.getSheets()[0]);
  for (var c = 2; c <= ss.getSheets()[2].getDataRange().getLastColumn(); c++){
    if (ss.getSheets()[2].getRange(1, c).getValue() == "SELECTED") {
      var languageSelected = c;
    }
  }
  app2.setTitle(ss.getSheets()[2].getRange(22,languageSelected).getValue());

  var panel = app2.createVerticalPanel();

  var label1 =
app2.createLabel(ss.getSheets()[2].getRange(23,languageSelected).getValue());

```

```

var text1 = app2.createTextBox();

text1.setName("TextBox1");

var button = app2.createButton("Ok");

panel.setSpacing(5);
panel.add(label1);
panel.add(text1);

panel.add(button);

var handler = app2.createServerHandler("onEditRow");

button.addClickHandler(handler);
handler.addCallbackElement(panel);

var scroll = app2.createScrollPane().setPixelSize(500, 300);
scroll.add(panel);
app2.add(scroll);

var doc = SpreadsheetApp.getActiveSpreadsheet();
doc.show(app2);
}

```

Aquest formulari ens demana el número d'identificador del client el qual volem modificar.

```

function onEditRow (e){

var app = UiApp.createApplication();
var num = e.parameter.TextBox1;
//var num = 134;
//in order to find row num entrie i create an array with all the numbers and extract the row

var ss = SpreadsheetApp.getActiveSpreadsheet();
ss.setActiveSheet(ss.getSheets()[0]);

var range = SpreadsheetApp.getActiveSheet().getDataRange();

var lastColumn = SpreadsheetApp.getActiveSheet().getLastColumn();
var range2 = SpreadsheetApp.getActiveSheet().getRange(1,1,1,lastColumn);
for (var x = 0; x < range2.getLastColumn(); x++) {
  if (range2.offset(0, x, 1, 1).getValue() == "Num client ") {
    var numClient = x;
  }
  if (range2.offset(0, x, 1, 1).getValue() == "Origen reserva ") {
    var origenCol = x;
  }
  if (range2.offset(0, x, 1, 1).getValue() == "Nom i cognom ") {

```

```

var nameAndSurnameCol = x;
}
if (range2.offset(0, x, 1, 1).getValue() == "Email client ") {
  var emailCol = x;
}
if (range2.offset(0, x, 1, 1).getValue() == "Dades de contacte (telfs, adreces,...) ") {
  var contactDataCol = x;
}
if (range2.offset(0, x, 1, 1).getValue() == "Data d'entrada ") {
  var dateInColumnCol = x;
}
if (range2.offset(0, x, 1, 1).getValue() == "Data de sortida ") {
  var dateOutColumnCol = x;
}
if (range2.offset(0, x, 1, 1).getValue() == "# dies ") {
  var numDaysCol = x;
}
if (range2.offset(0, x, 1, 1).getValue() == "# persones ") {
  var numPersonCol = x;
}
if (range2.offset(0, x, 1, 1).getValue() == "# adults ") {
  var numAdultsCol = x;
}
if (range2.offset(0, x, 1, 1).getValue() == "# nens ") {
  var numChildCol = x;
}
if (range2.offset(0, x, 1, 1).getValue() == "Pais ") {
  var countryCol = x;
}
if (range2.offset(0, x, 1, 1).getValue() == "Cost pel client ") {
  var costToTheCustomerCol = x;
}
if (range2.offset(0, x, 1, 1).getValue() == "Paga i senyal ") {
  var depositPaymentCol = x;
}
if (range2.offset(0, x, 1, 1).getValue() == "Saldo ") {
  var balanceCol = x;
}
if (range2.offset(0, x, 1, 1).getValue() == "Pagat ") {
  var paidCol = x;
}
if (range2.offset(0, x, 1, 1).getValue() == "Despeses de neteja ") {
  var cleaningExpenditureCol = x;
}
if (range2.offset(0, x, 1, 1).getValue() == "Despeses de gestió ") {
  var managementExpenditureCol = x;
}
if (range2.offset(0, x, 1, 1).getValue() == "Impostos ") {
  var managementTaxCol = x;
}
if (range2.offset(0, x, 1, 1).getValue() == "Guanys ") {

```

```

var managementGainCol = x;
}
if (range2.offset(0, x, 1, 1).getValue() == "Fiança ") {
  var managementFinancesCol = x;
}
if (range2.offset(0, x, 1, 1).getValue() == "Checklist ") {
  var managementChecklistCol = x;
}
if (range2.offset(0, x, 1, 1).getValue() == "Ressenya ") {
  var managementReviewCol = x;
}
if (range2.offset(0, x, 1, 1).getValue() == "Calendaris actualitzats ") {
  var calendarUpdatedCol = x;
}
if (range2.offset(0, x, 1, 1).getValue() == "Retorn fiança ") {
  var returnOfTheDepositCol = x;
}
if (range2.offset(0, x, 1, 1).getValue() == "Comentari ") {
  var commentsCol = x;
}
  if (range2.offset(0, x, 1, 1).getValue() == "Cancelacions") {
    var cancelCol = x;
  }
    if (range2.offset(0, x, 1, 1).getValue() == "Gestor Avisat") {
      var messageGCol = x;
    }
    if (range2.offset(0, x, 1, 1).getValue() == "Usuari Avisat") {
      var messageUCol = x;
    }
  }
}

var array = new Array();
for (var x = 1; x <= range.getLastRow(); x++) {
  if (num == range.getValues()[x][numClient]){
    var editedRow = x;
    var x = range.getLastRow();
  }
}
var originalDateIn =
app.createTextBox().setValue(range.getValues()[editedRow][dateInColumnCol]).setName("originalDateIn").setEnabled(false);
var originalDateOut =
app.createTextBox().setValue(range.getValues()[editedRow][dateOutColumnCol]).setName("originalDateOut").setEnabled(false);
var originalCancel =
app.createTextBox().setValue(range.getValues()[editedRow][cancelCol]).setName("originalCancel").setEnabled(false);
var originalBalance =
app.createTextBox().setValue(range.getValues()[editedRow][balanceCol]).setName("originalBalance").setEnabled(false);

```

```

var panel = app.createVerticalPanel();

var labelOportunity = app.createLabel("Oportunity name");
var textOportunity = app.createTextBox().setValue(range.getValues()[editedRow][0]);
var labelnum = app.createLabel("Número de client");
var num = app.createTextBox().setValue(range.getValues()[editedRow][numClient]);
var label1 = app.createLabel("Origen reserva");
var text1 = app.createTextBox().setValue(range.getValues()[editedRow][origenCol]);
var label2 = app.createLabel("Nom i cognom ");
var text2 =
app.createTextBox().setValue(range.getValues()[editedRow][nameAndSurnameCol]);
var label3 = app.createLabel("Email client ");
var text3 = app.createTextBox().setValue(range.getValues()[editedRow][emailCol]);
var label4 = app.createLabel("Dades de contacte (telfs, adreces,...) ");
var text4 = app.createTextBox().setValue(range.getValues()[editedRow][contactDataCol]);
var labeldate1 = app.createLabel("Data d'entrada");
var date1 =
app.createDateBox().setValue(range.getValues()[editedRow][dateInColumnCol]);
var labeldate2 = app.createLabel("Data de sortida");
var date2 =
app.createDateBox().setValue(range.getValues()[editedRow][dateOutColumnCol]);
// var label5 = app.createLabel("# dies");
// var text5 = app.createTextBox();
// var label6 = app.createLabel("# persones");
// var text6 = app.createTextBox();
var label7 = app.createLabel("# adults");
var text7 = app.createTextBox().setValue(range.getValues()[editedRow][numAdultsCol]);
var label8 = app.createLabel("# nens");
var text8 = app.createTextBox().setValue(range.getValues()[editedRow][numChildCol]);
var label9 = app.createLabel("Pais");
var text9 = app.createTextBox().setValue(range.getValues()[editedRow][countryCol]);
var label10 = app.createLabel("Cost pel client");
var text10 =
app.createTextBox().setValue(range.getValues()[editedRow][costToTheCustomerCol]);
var label11 = app.createLabel("Paga i senyal");
var text11 =
app.createTextBox().setValue(range.getValues()[editedRow][depositPaymentCol]);
var label12 = app.createLabel("Saldo");
// var text12 = app.createTextBox().setValue(range.getValues()[editedRow][balanceCol]);
var label13 = app.createLabel("Pagat");
var text13 = app.createTextBox().setValue(range.getValues()[editedRow][paidCol]);
var label14 = app.createLabel("Despeses de neteja");
var text14 =
app.createTextBox().setValue(range.getValues()[editedRow][cleaningExpenditureCol]);
var label15 = app.createLabel("Despeses de gestió");
var text15 =
app.createTextBox().setValue(range.getValues()[editedRow][managementExpenditureCol]);
var label16 = app.createLabel("Impostos");
var text16 =
app.createTextBox().setValue(range.getValues()[editedRow][managementTaxCol]);
// var label17 = app.createLabel("Guanys");

```

```

// var text17 =
app.createTextBox().setValue(range.getValues()[editedRow][managementGainCol]);
var label18 = app.createLabel("Fiança");
var text18 =
app.createTextBox().setValue(range.getValues()[editedRow][managementFinancesCol]);
var text19 =
app.createCheckBox("Checklist").setValue(range.getValues()[editedRow][managementChecklistCol]);
var text20 =
app.createCheckBox("Ressenya").setValue(range.getValues()[editedRow][managementReviewCol]);
var text21 = app.createCheckBox("Calendaris actualitzats").setValue(range.getValues()[editedRow][calendarUpdatedCol]);
var text22 = app.createCheckBox("Retorn fiança").setValue(range.getValues()[editedRow][returnOfTheDepositCol]);
var label23 = app.createLabel("Comentaris");
var text23 = app.createTextBox().setValue(range.getValues()[editedRow][commentsCol]);
var text24 =
app.createCheckBox("Cancel·lacions").setValue(range.getValues()[editedRow][cancelCol]);

textOportunity.setName("TextOportunity");
num.setName("NumClient");
text1.setName("TextBox1");
text2.setName("TextBox2");
text3.setName("TextBox3");
text4.setName("TextBox4");
date1.setName("Date1");
date2.setName("Date2");
// text5.setName("TextBox5");
// text6.setName("TextBox6");
text7.setName("TextBox7");
text8.setName("TextBox8");
text9.setName("TextBox9");
text10.setName("TextBox10");
text11.setName("TextBox11");
// text12.setName("TextBox12");
text13.setName("TextBox13");
text14.setName("TextBox14");
text15.setName("TextBox15");
text16.setName("TextBox16");
// text17.setName("TextBox17");
text18.setName("TextBox18");
text19.setName("TextBox19");
text20.setName("TextBox20");
text21.setName("TextBox21");
text22.setName("TextBox22");
text23.setName("TextBox23");
text24.setName("TextBox24");

var button = app.createButton("Ok");

```

```
panel.setSpacing(5);
panel.add(labelOportunity);
panel.add(textOportunity);
panel.add(labelnum);
panel.add(num);
panel.add(label1);
panel.add(text1);
panel.add(label2);
panel.add(text2);
panel.add(label3);
panel.add(text3);
panel.add(label4);
panel.add(text4);
panel.add(labeldate1);
panel.add(date1);
panel.add(originalDateIn);
panel.add(labeldate2);
panel.add(date2);
panel.add(originalDateOut);
// panel.add(label5);
// panel.add(text5);
// panel.add(label6);
// panel.add(text6);
panel.add(label7);
panel.add(text7);
panel.add(label8);
panel.add(text8);
panel.add(label9);
panel.add(text9);
panel.add(label10);
panel.add(text10);
panel.add(label11);
panel.add(text11);
panel.add(label12);
panel.add(originalBalance);
panel.add(label13);
panel.add(text13);
panel.add(label14);
panel.add(text14);
panel.add(label15);
panel.add(text15);
panel.add(label16);
panel.add(text16);
// panel.add(label17);
// panel.add(text17);
panel.add(label18);
panel.add(text18);
panel.add(text19);
panel.add(text20);
panel.add(text21);
panel.add(text22);
```

```

panel.add(label23);
panel.add(text23);
panel.add(text24);
panel.add(originalCancel);
panel.add(button);

var handler = app.createServerHandler("editButton");
button.addClickHandler(handler);
handler.addCallbackElement(panel);

var scroll = app.createScrollPane().setPixelSize(500, 300);
scroll.add(panel);
app.add(scroll);

var doc = SpreadsheetApp.getActiveSpreadsheet();
doc.show(app);
var calendarID = SpreadsheetApp.getActiveSpreadsheet().getSheets()[1].getSheetValues(1,
2, 1, 1).valueOf();
var cal = CalendarApp.getCalendarById(calendarID);
var event = cal.getEvents(range.getValues()[editedRow][dateInColumnCol],
range.getValues()[editedRow][dateOutColumnCol]);
if (event.length){
  for (var x = 0; x <= event.length; x++) {
    event[x].deleteEvent();
  }
}
doc.deleteRow(editedRow+1);
}

```

Carreguem les dades de la fila que correspon a un formulari molt semblant al del nou client i esborrem del full de càcul i del calendari les dades antigues.

Podrem observar algunes dades en format de “només lectura” per si volem fer modificacions en aquests camps.

```

function editButton (e){

  for (var c = 2; c <=
SpreadsheetApp.getActiveSpreadsheet().getSheets()[2].getDataRange().getLastColumn();
c++){
    if (SpreadsheetApp.getActiveSpreadsheet().getSheets()[2].getRange(1, c).getValue() ==
"SELECTED") {
      var languageSelected = c;
    }
}

var originalDateIn = e.parameter.originalDateIn;
var originalDateOut = e.parameter.originalDateOut;

```

```
var originalBalance = e.parameter.originalBalance;
var originalCancel = e.parameter.originalCancel;
var days = new Date(e.parameter.Date2 - e.parameter.Date1);
var dateTemp1 = e.parameter.Date1;
var date1 = new Date(dateTemp1.setHours(13, 00, 00));
var dateTemp2 = e.parameter.Date2;
var date2 = new Date(dateTemp2.setHours(11, 00, 00));
var people = +e.parameter.TextBox7 + +e.parameter.TextBox8;
var autoBalance = +e.parameter.TextBox10 - +e.parameter.TextBox11 -
+e.parameter.TextBox13;
var autoManagementGain = +e.parameter.TextBox10 - +e.parameter.TextBox14 -
+e.parameter.TextBox15 - +e.parameter.TextBox16;
var app = UiApp.getActiveApplication();

var textOportunity = e.parameter.TextOportunity;
var numValue = e.parameter.NumClient;
var textView = e.parameter.TextBox1;
var textView2 = e.parameter.TextBox2;
var textView3 = e.parameter.TextBox3;
var textView4 = e.parameter.TextBox4;
var dateValue = date1;
var dateValue2 = date2;
var textView5 = days.getDate();
var textView6 = people;
var textView7 = e.parameter.TextBox7;
var textView8 = e.parameter.TextBox8;
var textView9 = e.parameter.TextBox9;
var textView10 = e.parameter.TextBox10;
var textView11 = e.parameter.TextBox11;
var textView12 = autoBalance;
var textView13 = e.parameter.TextBox13;
var textView14 = e.parameter.TextBox14;
var textView15 = e.parameter.TextBox15;
var textView16 = e.parameter.TextBox16;
var textView17 = autoManagementGain;
var textView18 = e.parameter.TextBox18;
var textView19 = e.parameter.TextBox19;
var textView20 = e.parameter.TextBox20;
var textView21 = e.parameter.TextBox21;
var textView22 = e.parameter.TextBox22;
var textView23 = e.parameter.TextBox23;
var textView24 = e.parameter.TextBox24;
var textView25 = false;
var textView26 = false;

var sheet = SpreadsheetApp.getActiveSpreadsheet();
var lastCol = sheet.getSheets()[0].getLastColumn();
var lastRow = sheet.getSheets()[0].getLastRow();

var source = sheet.getSheets()[0]
var destination = sheet.getSheets()[0]
```

```

var range = source.getRange(lastRow, 1, 1, lastCol);

// This copies the formatting in last column and last row in the source sheet to
// D4:F6 in the same sheet
var newArray = new Array();
newArray = [textOpportunity, numValue, textView, textView2, textView3, textView4,
dateValue, dateValue2, textView5, textView6, textView7, textView8, textView9,
textView10, textView11, textView12, textView13, textView14, textView15, textView16,
textView17, textView18, textView19, textView20, textView21, textView22, textView23,
textView24, textView25, textView26];
sheet.appendRow(newArray);
range.copyFormatToRange(destination, 1, lastCol, lastRow+1, lastRow+1);
sheet.getSheets()[0].sort(7);//This code sort descendant by column 7
sheet.getSheets()[0].getRange(lastRow+1, 1, 1, lastCol).setBorder(true, true, true, true, true);

var today = new Date();

Browser.msgBox(SpreadsheetApp.getActiveSpreadsheet().getSheets()[2].getRange(31,languageSelected).getValue());

for (var x = 0; x <
SpreadsheetApp.getActiveSheet().getRange(1,1,1,SpreadsheetApp.getActiveSheet().getLastColumn()).getLastColumn(); x++) {
    if
(SpreadsheetApp.getActiveSheet().getRange(1,1,1,SpreadsheetApp.getActiveSheet().getLastColumn()).offset(0, x, 1, 1).getValue() == "Nom i cognom ") {
        var nameAndSurnameCol = x;
    }
    if
(SpreadsheetApp.getActiveSheet().getRange(1,1,1,SpreadsheetApp.getActiveSheet().getLastColumn()).offset(0, x, 1, 1).getValue() == "Data d'entrada ") {
        var dateInColumnCol = x;
    }
    if
(SpreadsheetApp.getActiveSheet().getRange(1,1,1,SpreadsheetApp.getActiveSheet().getLastColumn()).offset(0, x, 1, 1).getValue() == "# persones ") {
        var numPersonCol = x;
    }
    if
(SpreadsheetApp.getActiveSheet().getRange(1,1,1,SpreadsheetApp.getActiveSheet().getLastColumn()).offset(0, x, 1, 1).getValue() == "Cancelacions") {
        var cancelCol = x;
    }
}

if (originalBalance != autoBalance && autoBalance == 0){
    var emailAddress = "";
    for (var c = 2; c <=
SpreadsheetApp.getActiveSpreadsheet().getSheets()[1].getDataRange().getLastColumn();
c++){

```

```

emailAddress +=
SpreadsheetApp.getActiveSpreadsheet().getSheets()[1].getRange(2,c).getValue() + ", ";
} //end of constructing email list address to send notification that flat is paid
var subject =
SpreadsheetApp.getActiveSpreadsheet().getSheets()[2].getRange(25,languageSelected).get
Value() + textView2;
var message =
SpreadsheetApp.getActiveSpreadsheet().getSheets()[2].getRange(26,languageSelected).get
Value() + textView2;
MailApp.sendEmail(emailAddress, subject, message);
}

if ((originalDateIn.getMonth() == today.getMonth() && originalDateIn.getYear() ==
today.getYear() || originalDateOut.getMonth() == today.getMonth() &&
originalDateOut.getYear() == today.getYear()) && date1.getMonth() == today.getMonth() &&
date1.getYear() == today.getYear() && (date1 != originalDateIn || date2 != originalDateOut) ||

originalCancel != textView24){
    var report = "";
    var nextMonth = (today.getMonth()+1)%12; //isn't enought adding 1, because Month start
count since 0 to 11
    var modifiedEntryActualMonthRow = "";
    var modifiedEntryActualMonth = "";
    var actualMonthEntries = "";
    var emailAddress = "";
    var i = 1;
    var imin = 0;
    var imax = 0;

    do{
        if
(SpreadsheetApp.getActiveSpreadsheet().getSheets()[0].getDataRange().getValues()[i][dateIn
ColumnCol].getMonth() < today.getMonth() &&
SpreadsheetApp.getActiveSpreadsheet().getSheets()[0].getDataRange().getValues()[i][dateIn
ColumnCol].getYear() == today.getYear())){
            var imin = i;
        }
        if
(SpreadsheetApp.getActiveSpreadsheet().getSheets()[0].getDataRange().getValues()[i][dateIn
ColumnCol].getMonth() == today.getMonth() &&
SpreadsheetApp.getActiveSpreadsheet().getSheets()[0].getDataRange().getValues()[i][dateIn
ColumnCol].getYear() == today.getYear())){
            var imax = i;
        }
        i++;
    }while
(SpreadsheetApp.getActiveSpreadsheet().getSheets()[0].getDataRange().getValues()[i][dateIn
ColumnCol].getMonth() < nextMonth)
    var imin = imin+1;

    for (var i = imin; i <= imax; i++) {
        var actualMonthEntriesRows = "";

```

```

for (var e = nameAndSurnameCol; e <= numPersonCol; e++){
    actualMonthEntriesRows += "<td style='padding:5px>" +
SpreadsheetApp.getActiveSpreadsheet().getSheets()[0].getDataRange().getValues()[i][e] +
"</td>";
}
actualMonthEntries += "<tr>" + actualMonthEntriesRows + "</tr>";
}

for (var e = nameAndSurnameCol; e <= numPersonCol; e++){
    modifiedEntryActualMonthRow += "<td style='padding:5px>" + newArray[e] + "</td>";
}
modifiedEntryActualMonthRow += "<td style='padding:5px>" + newArray[cancelCol] +
"</td>";
modifiedEntryActualMonth += "<tr>" + modifiedEntryActualMonthRow + "</tr>";

report = "<head><style>table, th, td {border: 1px solid black; border-collapse: collapse;}th, td
{padding: 15px;}</style></head><table><tr><th>Nom i cognom</th><th>Email client</th><th>Dades de contacte (telfs, adreces,...)</th><th>Data d'entrada</th><th>Data de sortida</th><th># dies</th><th># persones</th></tr>" +
actualMonthEntries + "</table><h2>" +
SpreadsheetApp.getActiveSpreadsheet().getSheets()[2].getRange(24,languageSelected).getValue() + "</h2><head><style>table, th, td {border: 1px solid black; border-collapse: collapse;}th, td
{padding: 15px;}</style></head><table><tr><th>Nom i cognom</th><th>Email client</th><th>Dades de contacte (telfs, adreces,...)</th><th>Data d'entrada</th><th>Data de sortida</th><th># dies</th><th># persones</th><th>Cancel·lacions</th></tr>" + modifiedEntryActualMonth +
"</table></body></html>";

for (var c = 2; c <=
SpreadsheetApp.getActiveSpreadsheet().getSheets()[1].getDataRange().getLastColumn();
c++){
    emailAddress +=
SpreadsheetApp.getActiveSpreadsheet().getSheets()[1].getRange(3,c).getValue() + ", ";
} //end of constructing email list address to send calendar
var subject =
SpreadsheetApp.getActiveSpreadsheet().getSheets()[2].getRange(20,languageSelected).getValue();
MailApp.sendEmail(emailAddress, subject, report, {htmlBody: report});

} //end of if to create monthly actual report
Browser.msgBox("Ok!");
return app.close();
}

```

En fer clic a “ok” es guarden les noves dades al full de càcul.

L'script compara si el valor que hem modificat ha sigut el del pagament i si ara l'estada està pagada totalment (no estant pagada abans). De ser així, enviarà un mail al gestor informant-lo.

També comprova si s'ha cancel·lat la reserva i si les dates han estat modificades tant en el cas de que la reserva estigués prevista per una data fora del més actual i es modifica per a que estigui prevista dins del més actual o en cas contrari, és a dir, que estigués planificada per una data dintre del més actual i es modifiqués per una data fora del més actual. En qualsevol d'aquests casos s'enviarà un mail al personal de neteja amb el calendari del més actual actualitzat i amb la nova fila que ha editat aquest calendari per a que no hi hagi confusions. Aquesta fila tindrà una columna extra l'email: "Cancel·lacions". D'aquesta manera no hi haurà dubte del que ha passat amb la modificació si aquesta no apareix al calendari del més actual.

El pròxim cop que s'executi l'script s'inclourà al google calendar.

```
function menuitem3() {
  for (var c = 2; c <=
SpreadsheetApp.getActiveSpreadsheet().getSheets()[2].getDataRange().getLastColumn();
c++){
  if (SpreadsheetApp.getActiveSpreadsheet().getSheets()[2].getRange(1, c).getValue() ==
"SELECTED") {
    var languageSelected = c;
  }
}

var app2 = UiApp.createApplication();
app2.setTitle(SpreadsheetApp.getActiveSpreadsheet().getSheets()[2].getRange(32,language
Selected).getValue());
var panel = app2.createVerticalPanel();
var today = new Date();

var label1 =
app2.createLabel(SpreadsheetApp.getActiveSpreadsheet().getSheets()[2].getRange(33,langu
ageSelected).getValue());

var labeldate1 =
app2.createLabel(SpreadsheetApp.getActiveSpreadsheet().getSheets()[2].getRange(34,langu
ageSelected).getValue());
var date1 = app2.createDateBox().setValue(today);
var labeldate2 =
app2.createLabel(SpreadsheetApp.getActiveSpreadsheet().getSheets()[2].getRange(35,langu
ageSelected).getValue());
var date2 = app2.createDateBox().setValue(today);

date1.setName("Date1");
date2.setName("Date2");

var button = app2.createButton("Ok");

panel.setSpacing(5);
panel.add(label1);
panel.add(labeldate1);
panel.add(date1);
```

```

panel.add(labeldate2);
panel.add(date2);

panel.add(button);

var handler = app2.createServerHandler("reporting");

button.addClickHandler(handler);
handler.addCallbackElement(panel);

var scroll = app2.createScrollPane().setPixelSize(500, 300);
scroll.add(panel);
app2.add(scroll);

var doc = SpreadsheetApp.getActiveSpreadsheet();
doc.show(app2);
}

```

Aquí introduirem la data d'inici i la de finalització del resum econòmic que volem generar. El qual ens crearà una nova fulla al final de les que ja tenim amb el nom de “Resum...” i les dates del rang que hem definit.

```

function reporting (e){
  for (var c = 2; c <=
SpreadsheetApp.getActiveSpreadsheet().getSheets()[2].getDataRange().getLastColumn());
c++){
  if (SpreadsheetApp.getActiveSpreadsheet().getSheets()[2].getRange(1, c).getValue() ==
"SELECTED") {
    var languageSelected = c;
  }
}

var app2 = UiApp.getActiveApplication();

var startDateValue = e.parameter.Date1;
var endDateValue = e.parameter.Date2;
/*
var startDateValue = new Date (new Date().getTime() - 365 * 24 * 60 * 60 * 1000);
var endDateValue = new Date (new Date().getTime() + 100 * 24 * 60 * 60 * 1000);
*/
var ss = SpreadsheetApp.getActiveSpreadsheet();
ss.setActiveSheet(ss.getSheets()[0]);
var lastColumn = SpreadsheetApp.getActiveSheet().getLastColumn();
var range2 = SpreadsheetApp.getActiveSheet().getRange(1,1,1,lastColumn);

for (var x = 0; x < range2.getLastColumn(); x++) {
  if (range2.offset(0, x, 1, 1).getValue() == "Data d'entrada ") {
    var dateInColumnCol = x;

```

```

}
if (range2.offset(0, x, 1, 1).getValue() == "Data de sortida ") {
  var dateOutColumnCol = x;
}
if (range2.offset(0, x, 1, 1).getValue() == "Pagat ") {
  var paidCol = x;
}
if (range2.offset(0, x, 1, 1).getValue() == "Despeses de neteja ") {
  var cleaningExpenditureCol = x;
}
if (range2.offset(0, x, 1, 1).getValue() == "Despeses de gestió ") {
  var managementExpenditureCol = x;
}
if (range2.offset(0, x, 1, 1).getValue() == "Impostos ") {
  var managementTaxCol = x;
}
}

for (var f = 3; f >= 0; f -= 3){
  ss.setActiveSheet(ss.getSheets()[f]);
  ss.getSheets()[f].sort(dateInColumnCol+1); //This code sort descendent by arrival data
column. The +1 is for this function the first column is 1. But the function getRange the first
column is 0
  var range = SpreadsheetApp.getActiveSheet().getDataRange();

  for (var i = range.getRow(); i <
SpreadsheetApp.getActiveSpreadsheet().getSheets()[f].getLastRow(); i++) {
    var rowValues = new Array(range.getLastColumn());

    for (var x = 0; x < range.getLastColumn(); x++) {
      rowValues[x] = range.getValues()[i][x];
    }
    var rowRange = range.offset(i, 0, 1);
    var dateInColumn = range.getValues()[i][dateInColumnCol];
    var dateOutColumn = range.getValues()[i][dateOutColumnCol];
    var paid = range.getValues()[i][paidCol].toString().replace(".", ",");//convert to string,
replace comma with dot
    var cleaningExpenditure2 = range.getValues()[i][cleaningExpenditureCol];
    var cleaningExpenditure =
range.getValues()[i][cleaningExpenditureCol].toString().replace(".", ",");
    var cleaningExpenditure2 = Utilities.formatString('%11,2f', cleaningExpenditure2);
    var managementExpenditure =
range.getValues()[i][managementExpenditureCol].toString().replace(".", ",");
    var managementTax = range.getValues()[i][managementTaxCol];
    var today = new Date();

    var currentMonth = today.getMonth();
    var tomorrow = new Date(today.getTime() + 1 * 24 * 60 * 60 * 1000);
    var lastyear = new Date(today.getTime() + 365 * 24 * 60 * 60 * 1000);

    var yearToCompare = today.getYear();
  }
}

```

```

var actualYear = today.getYear();
var yearAndMonthLastRow = "";
var anualReporting = "";
var mensualReportingIncome = "";
var mensualReportingExpense = "";
var mensualReportingGain = "";
var anualReportingIncome = "";
var anualReportingExpense = "";
var anualReportingGain = "";
var twelveLastMonthReportingIncome = "=";
var twelveLastMonthReportingExpense = "=";
var twelveLastMonthReportingGain = "";
var monthsNames = [['January', 'February', 'March', 'April', 'May', 'June', 'July', 'August',
'September', 'October', 'November', 'December'], ['Enero', 'Febrero', 'Marzo', 'Abril', 'Mayo',
'Junio', 'Julio', 'Agosto', 'Septiembre', 'Octubre', 'Noviembre', 'Diciembre'], ['Gener', 'Febrer',
'Març', 'Abril', 'Maig', 'Juny', 'Juliol', 'Agost', 'Setembre', 'Octubre', 'Novembre', 'Decembre']];
var yearAndMonthActualRow = monthsNames[languageSelected-
2][dateInColumn.getMonth()] + " " + dateInColumn.getYear();
//var yearAndMonthActualRow = Utilities.formatDate(dateInColumn, "GMT", "MMMM
yyyy");
var findMY = false;
var findY = false;
var findLastYear = false;
var numOfSheets = ss.getSheets().length;
var sheetName =
ss.getSheetByName(SpreadsheetApp.getActiveSpreadsheet().getSheets()[2].getRange(36,la
nguageSelected).getValue() + " " + startDateValue.toDateString() + " " +
SpreadsheetApp.getActiveSpreadsheet().getSheets()[2].getRange(37,languageSelected).get
Value() + " " + endDateValue.toDateString());
if (startDateValue <= dateInColumn && endDateValue >= dateOutColumn){ //miramos si
la fecha está en el rango que hemos pedido
  if (sheetName == null) { // si la nueva hoja no está creada, la creamos con el nombre de
report y las fechas escogidas
    var numOfSheetsPlusOne = numOfSheets+1;
    ss.insertSheet(SpreadsheetApp.getActiveSpreadsheet().getSheets()[2].getRange(36,languag
eSelected).getValue() + " " + startDateValue.toDateString() + " " +
SpreadsheetApp.getActiveSpreadsheet().getSheets()[2].getRange(37,languageSelected).get
Value() + " " + endDateValue.toDateString(), numOfSheetsPlusOne);
    ss.getSheets()[numOfSheetsPlusOne-1].getRange(2,
2).setValue(SpreadsheetApp.getActiveSpreadsheet().getSheets()[2].getRange(38,languageS
elected).getValue()).setBorder(true, true, true, true, true, true);
    ss.getSheets()[numOfSheetsPlusOne-1].getRange(3,
2).setValue(SpreadsheetApp.getActiveSpreadsheet().getSheets()[2].getRange(39,languageS
elected).getValue()).setBorder(true, true, true, true, true, true);
    ss.getSheets()[numOfSheetsPlusOne-1].getRange(4,
2).setValue(SpreadsheetApp.getActiveSpreadsheet().getSheets()[2].getRange(40,languageS
elected).getValue()).setBorder(true, true, true, true, true, true);
    ss.getSheets()[numOfSheetsPlusOne-1].getRange(5,
2).setValue(SpreadsheetApp.getActiveSpreadsheet().getSheets()[2].getRange(41,languageS
elected).getValue());
  }
}

```

```

elected).getValue()).setBorder(true, true, true, true, true, true);

    ss.getSheets()[numOfSheetsPlusOne-1].getRange(7,
2).setValue(SpreadsheetApp.getActiveSpreadsheet().getSheets()[2].getRange(38,languageS
elected).getValue()).setBorder(true, true, true, true, true, true);
    ss.getSheets()[numOfSheetsPlusOne-1].getRange(8,
2).setValue(SpreadsheetApp.getActiveSpreadsheet().getSheets()[2].getRange(39,languageS
elected).getValue()).setBorder(true, true, true, true, true, true);
    ss.getSheets()[numOfSheetsPlusOne-1].getRange(9,
2).setValue(SpreadsheetApp.getActiveSpreadsheet().getSheets()[2].getRange(40,languageS
elected).getValue()).setBorder(true, true, true, true, true, true);
    ss.getSheets()[numOfSheetsPlusOne-1].getRange(10,
2).setValue(SpreadsheetApp.getActiveSpreadsheet().getSheets()[2].getRange(41,languageS
elected).getValue()).setBorder(true, true, true, true, true, true);
}

for (var columnEconomic = 3; columnEconomic <=
SpreadsheetApp.getActiveSpreadsheet().getSheets()[numOfSheetsPlusOne-
1].getLastColumn() + 1; columnEconomic++) { // comprueba si en la primera columna vacía está
el mes y años que estamos mirando en la fila actual
    if (yearAndMonthActualRow ==
SpreadsheetApp.getActiveSpreadsheet().getSheets()[numOfSheetsPlusOne-
1].getSheetValues(2, columnEconomic, 1, 1) && findMY == false){
        var econoCol = columnEconomic;
        var findMY = true;
    }
}

for (var columnEconomic = 3; columnEconomic <=
SpreadsheetApp.getActiveSpreadsheet().getSheets()[numOfSheetsPlusOne-
1].getLastColumn() + 1; columnEconomic++) { // comprueba si es la primera vez y está
vacío, de ser así lo crea
    if (SpreadsheetApp.getActiveSpreadsheet().getSheets()[numOfSheetsPlusOne-
1].getSheetValues(2, columnEconomic, 1, 1) == "" && findMY == false){
        var econoCol = columnEconomic;
        ss.getSheets()[numOfSheetsPlusOne-1].getRange(2,
econoCol).setValue(yearAndMonthActualRow).setBorder(true, true, true, true, true, true);
        ss.getSheets()[numOfSheetsPlusOne-1].getRange(3,
econoCol).setValue("=").setBorder(true, true, true, true, true, true);
        ss.getSheets()[numOfSheetsPlusOne-1].getRange(4,
econoCol).setValue("=").setBorder(true, true, true, true, true, true);
        ss.getSheets()[numOfSheetsPlusOne-1].getRange(5,
econoCol).setValue("=").setBorder(true, true, true, true, true, true);
        var findMY = true;
    }
}

mensualReportingIncome = ss.getSheets()[numOfSheetsPlusOne-1].getRange(3,
econoCol).getFormula();
mensualReportingExpense = ss.getSheets()[numOfSheetsPlusOne-1].getRange(4,
econoCol).getFormula();

```

```

mensualReportingGain = ss.getSheets()[numOfSheetsPlusOne-1].getRange(5,
econoCol).getFormula();
ss.getSheets()[numOfSheetsPlusOne-1].getRange(3,
econoCol).setValue(mensualReportingIncome + "+" + paid).setBorder(true, true, true, true,
true, true);
ss.getSheets()[numOfSheetsPlusOne-1].getRange(4,
econoCol).setValue(mensualReportingExpense + "+" + cleaningExpenditure + "+" +
managementExpenditure + "+" + managementTax).setBorder(true, true, true, true, true, true);
ss.getSheets()[numOfSheetsPlusOne-1].getRange(5,
econoCol).setValue(mensualReportingGain + "+" + paid + "-" + cleaningExpenditure + "-" +
managementExpenditure + "-" + managementTax).setBorder(true, true, true, true, true, true);

for (var columnEconomic = 3; columnEconomic <=
SpreadsheetApp.getActiveSpreadsheet().getSheets()[numOfSheetsPlusOne-
1].getLastColumn() + 1; columnEconomic++) {
    if (dateInColumn.getYear() ==
SpreadsheetApp.getActiveSpreadsheet().getSheets()[numOfSheetsPlusOne-
1].getSheetValues(7, columnEconomic, 1, 1) && findY == false) {
        var econoCol = columnEconomic;
        var findY = true;
    }
}

for (var columnEconomic = 3; columnEconomic <=
SpreadsheetApp.getActiveSpreadsheet().getSheets()[numOfSheetsPlusOne-
1].getLastColumn() + 1; columnEconomic++) {
    if (SpreadsheetApp.getActiveSpreadsheet().getSheets()[numOfSheetsPlusOne-
1].getSheetValues(7, columnEconomic, 1, 1) == "" && findY == false) {
        var econoCol = columnEconomic;
        ss.getSheets()[numOfSheetsPlusOne-1].getRange(7,
econoCol).setValue(dateInColumn.getYear()).setBorder(true, true, true, true, true, true);
        ss.getSheets()[numOfSheetsPlusOne-1].getRange(8,
econoCol).setValue("=").setBorder(true, true, true, true, true, true);
        ss.getSheets()[numOfSheetsPlusOne-1].getRange(9,
econoCol).setValue("=").setBorder(true, true, true, true, true, true);
        ss.getSheets()[numOfSheetsPlusOne-1].getRange(10,
econoCol).setValue("=").setBorder(true, true, true, true, true, true);
        var findY = true;
    }
}

anualReportingIncome = ss.getSheets()[numOfSheetsPlusOne-1].getRange(8,
econoCol).getFormula();
anualReportingExpense = ss.getSheets()[numOfSheetsPlusOne-1].getRange(9,
econoCol).getFormula();
anualReportingGain = ss.getSheets()[numOfSheetsPlusOne-1].getRange(10,
econoCol).getFormula();
ss.getSheets()[numOfSheetsPlusOne-1].getRange(8,
econoCol).setValue(anualReportingIncome + "+" + paid).setBorder(true, true, true, true, true,
true);

```

```

ss.getSheets()[numOfSheetsPlusOne-1].getRange(9,
econoCol).setValue(anualReportingExpense + "+" + cleaningExpenditure + "+" +
managementExpenditure + "+" + managementTax).setBorder(true, true, true, true, true);
ss.getSheets()[numOfSheetsPlusOne-1].getRange(10,
econoCol).setValue(anualReportingGain + "+" + paid + "-" + cleaningExpenditure + "-" +
managementExpenditure + "-" + managementTax).setBorder(true, true, true, true, true);
}
}
}//end of economic block
Browser.msgBox("Ok!");
return app2.close();
}

```

Si volem un altre resum en el mateix rang perquè hem modificat alguna dada caldrà esborrar aquesta fulla i generar-la de nou. El resum accepta decimals i selecció d'idioma.

6.3 MODELS DE MISSATGES DE L'APLICACIÓ

En aquesta secció trobem una mostra dels models de correus electrònic que l'aplicació fa arribar al gestor i el personal a partir dels events anteriorment descrits.

6.2.1 AVISOS AL GESTOR



Fig. 17. Avís de no pagament

Avis de solapament Inbox x

 oliva.albert@gmail.com
to me 2:17 AM (0 minutes ago) ☆

Catalan ▾ > English ▾ Translate message Turn off for: Catalan x

L'estada de Leeanne Kepley colisiona. Si us plau revisar.

 oliva.albert@gmail.com
to me 2:17 AM (0 minutes ago) ☆

Catalan ▾ > English ▾ Translate message Turn off for: Catalan x

L'estada de Jesus Alcober colisiona. Si us plau revisar.

 oliva.albert@gmail.com
to me 2:17 AM (0 minutes ago) ☆

Catalan ▾ > English ▾ Translate message Turn off for: Catalan x

L'estada de Modesta Murrah colisiona. Si us plau revisar.

 oliva.albert@gmail.com
to me 2:17 AM (0 minutes ago) ☆

Catalan ▾ > English ▾ Translate message Turn off for: Catalan x

L'estada de Kera Collins colisiona. Si us plau revisar.

 oliva.albert@gmail.com
to me 2:17 AM (0 minutes ago) ☆

Catalan ▾ > English ▾ Translate message Turn off for: Catalan x

L'estada de Warner Luckett colisiona. Si us plau revisar.

Fig.18.Avis de solapament

6.2.2 AVISOS AL PERSONAL DE MANTENIMENT

Mover a Recibidos Más ▾

Resum mensual Recibidos x

 oliva.albert@gmail.com
para mí 7 de abr. (hace 13 días) ☆

Nom i cognom	Email client	Dades de contacte (telfs, adrescs,...)	Data d'entrada	Data de sortida	# dies	# persones
Antoinette Finch	client09@aaaida.com		Sun May 03 2015 13:00:00 GMT+0200 (CEST)	Wed May 06 2015 11:00:00 GMT+0200 (CEST)	3	4
Alysha Schacht	client10@aaaida.com		Sun May 10 2015 13:00:00 GMT+0200 (CEST)	Fri May 15 2015 11:00:00 GMT+0200 (CEST)	5	4
Sharita Bock	client11@aaaida.com		Mon May 18 2015 13:00:00 GMT+0200 (CEST)	Mon Jun 01 2015 11:00:00 GMT+0200 (CEST)	14	4

 Haz clic aquí si quieras [Responder](#) o [Reenviar](#) el mensaje

Fig.19.Resum mensual

6.3 BIBLIOGRAFIA CONSULTADA

- TUTORIAL Gestión de proyectos ágiles con Pivotal Tracker. Alejandro Pérez García.
<http://www.adictosaltrabajo.com/tutoriales/tutoriales.php?pagina=pivotalTracker>
- Documentació oficial de Google SpreadSheet Service per a desenvolupadors.
<https://developers.google.com/apps-script/reference/spreadsheet/>