Functional specifications of Tuten Solution

Version 1.2 – March 2018

Changes history

Required by	Roi Amszynowski
Date	01/01/2017
Detail	Original Document Creation
Attended by	Andrés Araya
Version	1.0

Required by	Roi Amszynowski
Date	03/01/2018
Detail	Document Redesign
Attended by	Livio Carrasco
Version	1.2

Required by	Roi Amszynowski
Date	03/14/2018
Details	Application of Improvements to text.
Attended by	Livio Carrasco
Version	1.27

CONTENTS

WHAT IS TUTEN

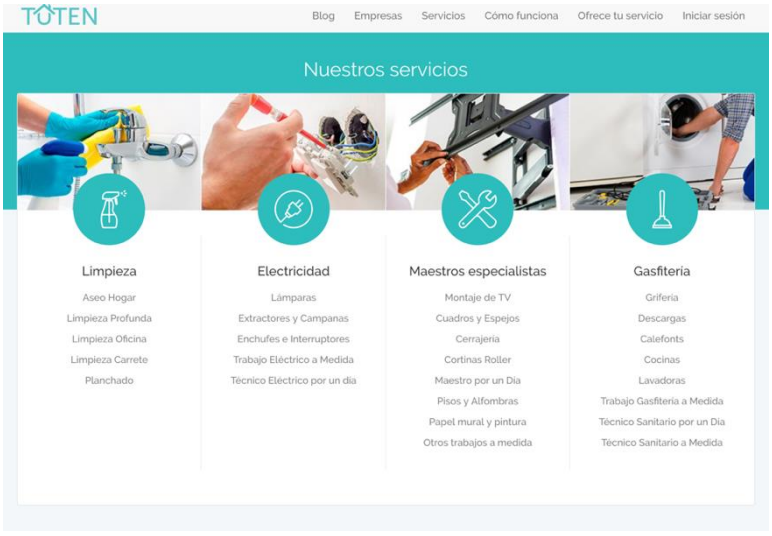
Tuten is a company that offers home integral management of services, such as home cleaning services, electricians, gas installers, and locksmiths, among others. Within this management, we offer a modular system, which allows you to schedule, pay and manage all services online in Chile and Latin America. To do this, Tuten offers a platform, which can be accessed online, using both desktop web browser and mobile devices.
Using *Complex Event Processing Technology*, various algorithms and technologies developed by our company, an appropriate service professional is assigned to the job, guaranteeing the professionals presence to perform the job on a date and place agreed, in accordance with to Tuten's service standards.

Tuten's business model considers two broad areas:

- B2C Services: Under this modality, physical services are offered to the final customer directly by Tuten.
- B2B Services: Under this modality, Tuten offers both its network of service providers and its technological platform in the private label modality, in order for other companies to be able to offer services to their clients.

The provision of physical services is based on the *On-Demand* business model. Independent professionals register themselves to provide services based on their availability, and the TUTEN Platform sends them specific job offers they must accept in order to perform the service. All service professionals registered in the Tuten Platform undergo a strict selection, training and supervision process to guarantee the quality of our work.

Tuten differentiates itself from its competitors because it offers an automated work management and assignment platform with high control over all operations, as well as having a business model that distinguishes it from its competitors, who have only focused on the B2C business model using manual processes.



Additionally, by offering the security of having certified contractors, Tuten also offers a guarantee on the jobs carried out by the contractor, including the refund of money among its policies in

cases where the work is not properly done, thus positioning Tuten as a certified, secure and trustworthy service.

TUTEN SOLUTION

Tuten is a company that offers home services through a transactional platform.

Based on the operation model, the mechanics of each operation and interaction are different. In them, more or less actors may intervene depending on the model.

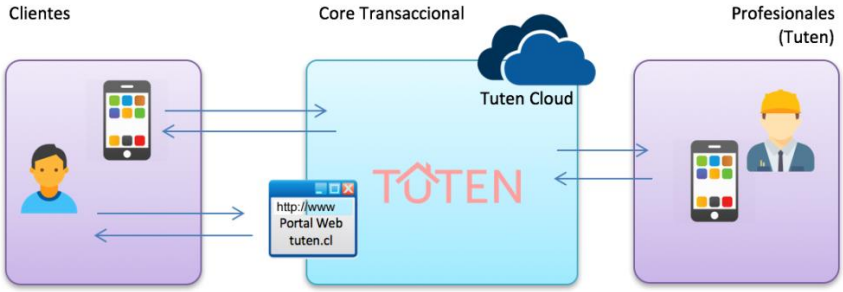
The most common solution models Tuten uses to operate are:

1. Traditional Business-to-Consumer Model (Traditional B2C)
2. Business-to-Business model operated by Tuten (B2B Suppliers network)
3. Business-to-Business model managed by other companies (B2B SaaS)

1.- Business-to-Consumer Model

This is Tuten's initial model, where Tuten manages the complete operational process.

In this context, there are 3 major actors which intervene in the process:



- ✓ **Customers**
  To whom Tuten offers different services. These communicate with Tuten through different devices, such as Mobile Apps (for IOS and Android) and web browsers. These requests are redirected to the Tuten's Transactional Core, which manages them.

- ✓ **Transactional Core**
  It is the infrastructure responsible for managing requests from customers or other participants in the process. It allows the scheduling services and the managing of tasks, professionals, customers, agendas, tickets, and all the necessary information for the business' proper functioning. It also allows the processing of notifications to professionals and/or companies using these services. Additionally, it allows the management of payments for services, as well as the control of information for the purposes of accounting and legal audits.

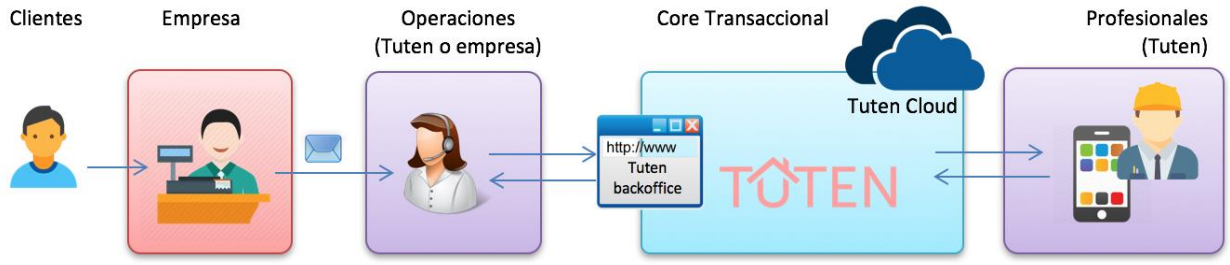
  All information going through the platform is stored in databases, where the corresponding *end-to-end* transaction monitoring and control is performed.

- ✓ **Professionals**
  They are the participants who provide the service offered by Tuten. These professionals work for Tuten (as a pool of professionals and/or subcontractors) in an *On-Demand* modality, in which Tuten manages their agendas of scheduled jobs. These professionals use a Proprietary App, through which they receive notifications regarding the scheduled jobs via *push* messages. On this App, the professional can review jobs in progress, previously done jobs, he/she can configure his/her availability, manage his/her own jobs (accept them or not), see the job's location, fill out an activities checklist, evaluate the feasibility of the job, add comments, attach evidence of job done, and other tasks.

2.- Business-to-Business Model Operated by Tuten

In this model, a company makes service requests directly to Tuten. In this modality, Tuten manages the complete operational process, but the presentation of requests can be made directly by the company, or it can deliver requests from their own customers to Tuten.

In this context, there are 4 major participants that intervene in the process:



- ✓ **Customers**
  Those to whom the company offers services. The client approaches the company to request a service, and registers all the necessary data to be able to schedule the service.

- ✓ **Company**
  The company takes the service request from the client and must proceed to schedule the service on the Tuten platform. There are two variants in this modality:

  1) The company enters the request directly on the Tuten's *backoffice*.
     In these cases, Tuten provides systems with *backend* access, which allows the company to carry out a comprehensive operational monitoring, which delivers sales, customers, and payment control information, among others, which support the company's operational and commercial management.

  2) The company delivers the request to Tuten by some channel, and Tuten's *staff* will enter the requests.

  In both cases, these requests are redirected to Tuten's Transactional Core, which manages them.

- ✓ **Transactional Core**
  It is the infrastructure responsible for managing requests from customers or other participants in the process. It allows the scheduling services and the managing of tasks, professionals, customers, agendas, tickets, and all the necessary information for the business' proper functioning. It also allows the processing of notifications to professionals and/or companies using these services. Additionally, it allows the management of

payments for services, as well as the control of information for the purposes of accounting and legal audits.

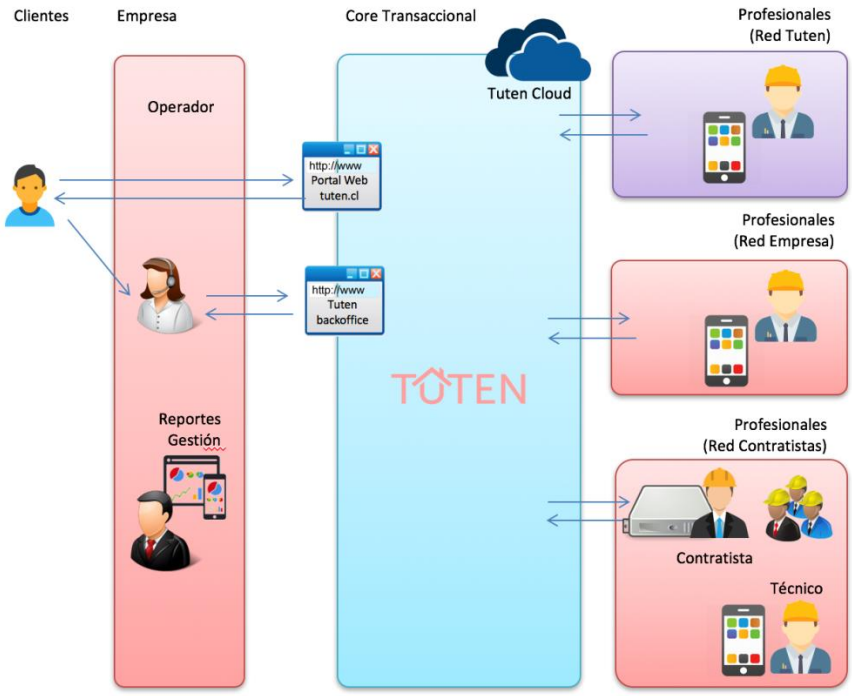
All information going through the platform is stored in databases, where the corresponding *end-to-end* transaction monitoring and control is performed.

- ✓ **Professionals**
  They are the participants who provide the service offered by Tuten. These professionals work for Tuten (as a pool of professionals and/or subcontractors) in an *On-Demand* modality, in which Tuten manages their agendas of scheduled jobs. These professionals use a Proprietary App, through which they receive notifications regarding the scheduled jobs via *push* messages. On this App, the professional can review jobs in progress, previously done jobs, he/she can configure his/her availability, manage his/her own jobs (accept them or not), see the job's location, fill out an activities checklist, evaluate the feasibility of the job, add comments, attach evidence of job done, and other tasks.

3.- Business-to-Business Model Administered By Companies (B2B SaaS)

In this model, a company wishes to make service requests directly to Tuten. In this modality, the company manages the entire operational process and delivers the requests made by its own customers through the Tuten platform. It is possible for Tuten to design specific components requested by each company, which causes this method to be known as SaaS (*Software-as-a-Service*).

In this context, four major participants are part of the process:



- ✓ **Customer**
  Those to whom the company offers services. The client approaches the company to request a service, and registers all the necessary data to be able to schedule the service.

- ✓ **Companies**
  The company takes the services request from the client and must proceed to schedule the service on the Tuten platform using Tuten's *backoffice*, where the company performs an integral operational follow-up, delivering sales, customers, and payment control information, among others services supporting the company's operational and commercial management.

  These requests are redirected to Tuten's Transactional Core, which manages them.

- ✓ **Transactional Core**

  It is the infrastructure responsible for managing requests from customers or other participants in the process. It allows the scheduling services and the managing of tasks, professionals, customers, agendas, tickets, and all the necessary information for the business' proper functioning. It also allows the processing of notifications to professionals and/or companies using these services. Additionally, it allows the management of payments for services, as well as the control of information for the purposes of accounting and legal audits.

  All information going through the platform is stored in databases, where the corresponding *end-to-end* transaction monitoring and control is performed.

- ✓ **Professionals**

  Those who provide the service offered by the company.
  There are several provider's modalities, based on the kind of professional who must provide the service, as well as on who it depends on organizationally.

  1) The professional depends on Tuten
     In this modality, the professionals work for Tuten (as a pool of professionals and/or subcontractors) in accordance with the *On-Demand* modality (Tuten manages their work schedule agendas).

  2) The professional works for the company
     In this modality, the professionals depend on the company which enters a scheduling request. They work for the company directly; however, their agendas are managed through the Tuten Platform.

     The company must provide a list of workers working under this modality to enter their information directly into its database.

  3) The professional belongs to a contractor company
     In this modality, the professionals work for a contractor company associated with the company presenting a scheduling request. They work directly for the contractor company; however, their agendas are managed through the Tuten Platform.

     The company and/or the contractor must provide a list of workers working under this modality to enter their information directly into its database.

  In any case, these professionals use a Proprietary App, through which they receive notifications regarding the scheduled jobs via *push* messages. On this App, the professional can review jobs in progress, previously done jobs, he/she can configure his/her availability, manage his/her own jobs (accept them or not), see the job's location,

fill out an activities checklist, evaluate the feasibility of the job, add comments, attach evidence of job done, and other tasks.

Additionally, Tuten offers scheduling notification systems (based on each company's technological capabilities), both for companies and for contractors, for them to know their agenda's status in real time.

OPERATION MODEL

Now that we are familiar with each model's singularities (B2C, B2B operated by Tuten, and B2B SaaS), a simplified diagram of the operation model specifically focused on the scheduling request process, from the moment where the application is entered and until the end of the process, is presented.

1.- Scheduling request:

Using the Tuten web portal or the Smartphone's app available on Apple store and Google Play (B2C modality), or through the company managing the service (B2B modality), the client indicates the type of service wanted and the date and time to carry it out. The payment method is also managed here (for B2C modalities, a credit or debit card via Transbank or Paypal can be used, while there may be other payment alternatives in accordance with the needs of each company in B2B modalities).

The request is received and managed in the company's *backend*.

2.- Scheduling Engine:

The Tuten engine selects the most suitable professionals to offer the job in real time, based on a set of predetermined rules, as well as other elements such as georeferencing conditions, kind of work, characteristics of the professional (new professionals, experienced professionals, etc.), temporary rules. There also exist definitions based on geolocalization conditions (country, city, etc.) that could eventually be considered part of the algorithm.

These rules are defined by the company, so each company can have its own set of rules.

3.- Scheduling:

The Tuten Platform sends a push message to the selected professionals pool. The first professional to accept the job will be the one to be present on the day and time requested by the client to perform the job. In addition to this, the companies using the service, either at professional and/or contracted services level, are notified. This is done regardless of whether the professional works for Tuten or whether he/she works for other companies and/or contractors.

Tuten has a scheduling notification systems, both for companies and contractors (based on each company's technological capabilities), for these companies to be able to know the status of their agendas in real time.

4.- Payment:

Once the job to be done is accepted, the services will be charged, if applicable.

In case the payment is appropriate, the Tuten platform generates the corresponding charges and is responsible for managing the charges involved, while those issuing commercial cards, or banks and banking entities are notified of the corresponding service charge.

In the B2B Services modality, there are other cases in which the payment is applied beforehand, for example, pre-paid charges, credit lines, monthly *fee* payments or others.

There are recurring services (for example, weekly cleaning services, where professional services are scheduled weekly in an automatic way). In these cases, the service is charged when it is executed.

4.- Notification to the professional/company

When the scheduling has already been completed, the professional receives a notification via push message through the Tuten Professionals' App with information about the job scheduled.

The professional can accept or reject the scheduled job. This will trigger a series of events:

- In the event the job is accepted, the App notifies the system that the job has been scheduled. In this case, the system notifies the client that the job has been accepted, and provides him/her information regarding the professional who will perform the job.

- In the event the professional does not accept the job, the App notifies the system, so that the platform can reassign the job to another professional qualified to perform the job.

In the B2B modality, both the acceptance and rejection of the job are notified to the company and/or contractor for whom the worker works, provided that the company and/or contractor has the appropriate technological systems for notifications of this kind of events.

5.- Execution of the Service

When the professional has accepted a job, he/she must provide follow up information regarding the execution of the same through a series of actions that must be performed on the mobile App.

First, the professional should begin the service, review the service instructions, complete the associated checklists, take pictures of the place, create additional services' budgets, etc.

Sometimes, it is possible that the job cannot be carried out, due to reasons such as product failure, delay in product delivery, place is not appropriate for the job, no agreement with the client, etc. In these cases, the service must be re-scheduled (marked as not possible to be carried out, and the Backoffice will schedule a new date).

In the event that the service requires additional work, an additional budget is generated through the professional's App (or through Tuten's *Backoffice*), which must be authorized by the client and the coordinators.

In cases where the service can be done correctly, the worker shall upload comments and photographs of the execution, as a complement to the work done.

All values regarding rejection reasons are measurable and are defined in accordance with business rules and/or the kind of service.

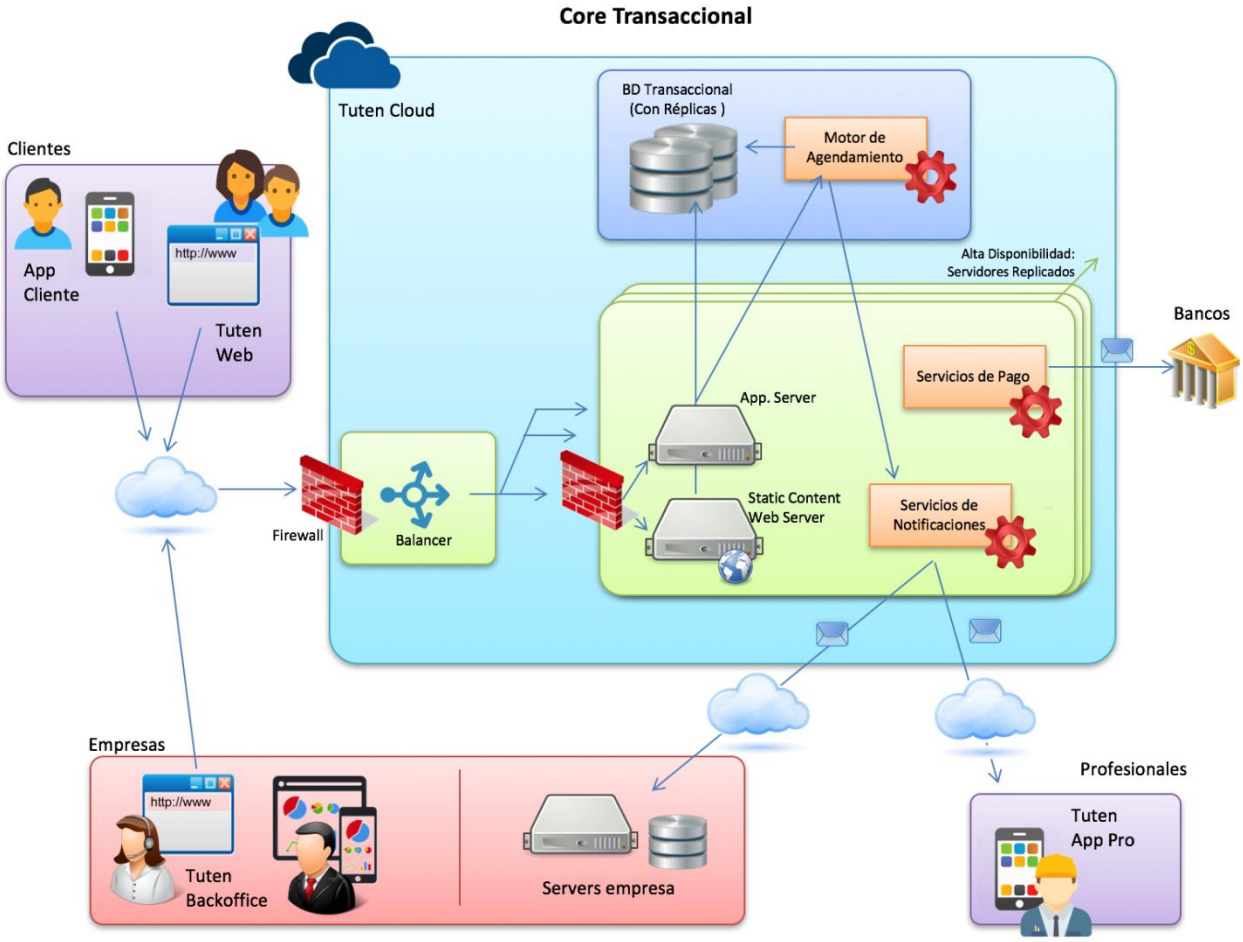
6.- Service rating

Once the service has been completed, the client will provide feedback on the job done by means of the client application and/or via email.

TUTEN SOLUTION INFRASTRUCTURE

Tuten's model is based on a robust platform called Transactional Core. This platform is hosted in a services cloud, which hosts several applications and servers supporting the totality of Tuten Solution processes.

In the following figure, it is possible to observe a simplified diagram of the different components and their interaction between them.

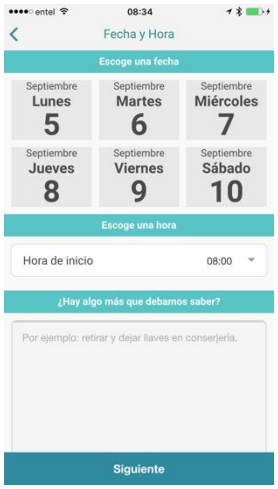


Some of the platforms' components will be briefly explained below.

CLIENT APP

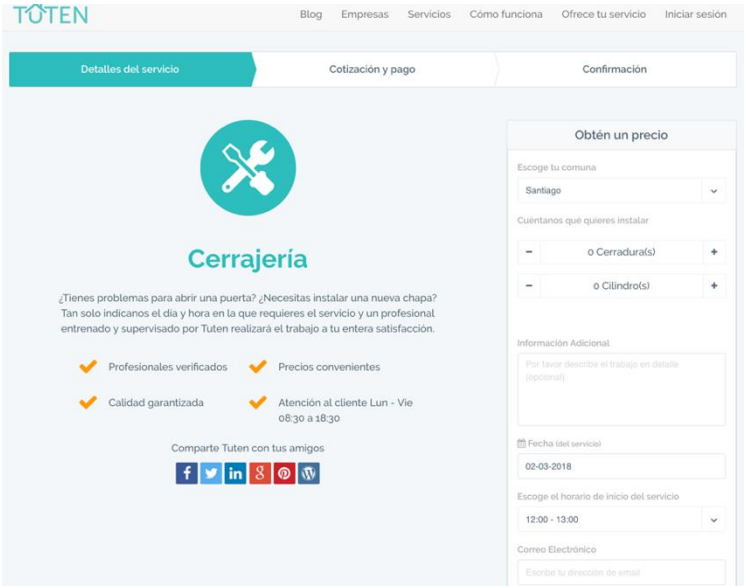
This is the Tuten's customers app, which can be used on a Smartphone. This App is available for both IOS and Android.

Using this App, the client can request and schedule services, such as cleaning, plumbing, electricity, locksmith services, etc. It also allows customers to verify scheduled jobs, cancel jobs in progress, check previously done jobs, and evaluate the services provided by the professionals, among other functionalities.



TUTEN WEB

It is a customer's website available at www.tuten.cl, which allows the customer to perform the same activities the Client App offers, that is, request and schedule services, such as cleaning, plumbing, electricity, locksmith services, etc. It also allows customers to verify scheduled jobs, cancel jobs in progress, check previously done jobs, and evaluate the services provided by the professionals, among other functionalities.



Additionally, this web platform also allows the uploading of professionals' applications to the Tuten's system by means of a predefined form.

This website can be customized in the event a company requests this service (B2B modalities), or requests additional components (B2B SaaS modality)

PROFESSIONALS APP (APP PRO)

This is the app for Tuten Professionals, which can be used on a Smartphone. It is available both for iOS and Android.

Through this app, the professional working with Tuten receives the jobs that the platform decides must be done by this professional (based on defined business rules).

The professional will receive a notification on his cell phone when he/she receives a job. Upon opening the app, a message with all the information related to the offered job will be displayed: kind of work, time needed, amount to be paid, job address (displayed on Google maps), customer's general information, and others.

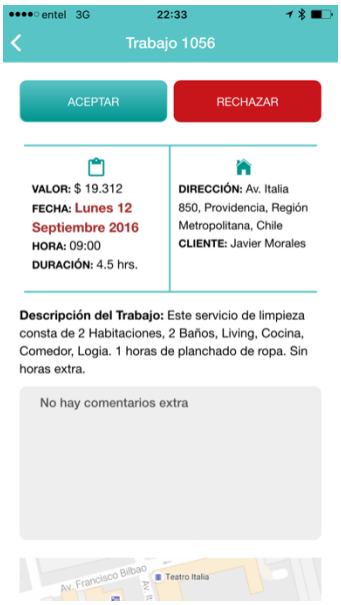
As these kinds of jobs are presented on the *On-Demand* modality, the professional can accept, reject or ignore the job.

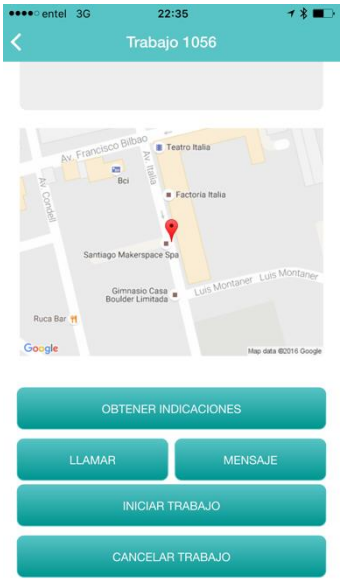
Once the job has been accepted, the professional must follow up on the completion of the same through the App, indicating the beginning of the job, filling in the tasks checklist, and formally reporting on the execution of the service.

In the event that the service requires additional work, an additional budget must be generated through this App (or by Tuten's *Backoffice*), which both by the client and the operational coordinators must authorized.

On the App, the professional can also upload comments and photos regarding the execution of the service, as a complement to the job done.

In addition, the app allows him/her to review jobs in progress and previously completed jobs, as well as to define an agenda of available dates and times.





TUTEN BACKOFFICE

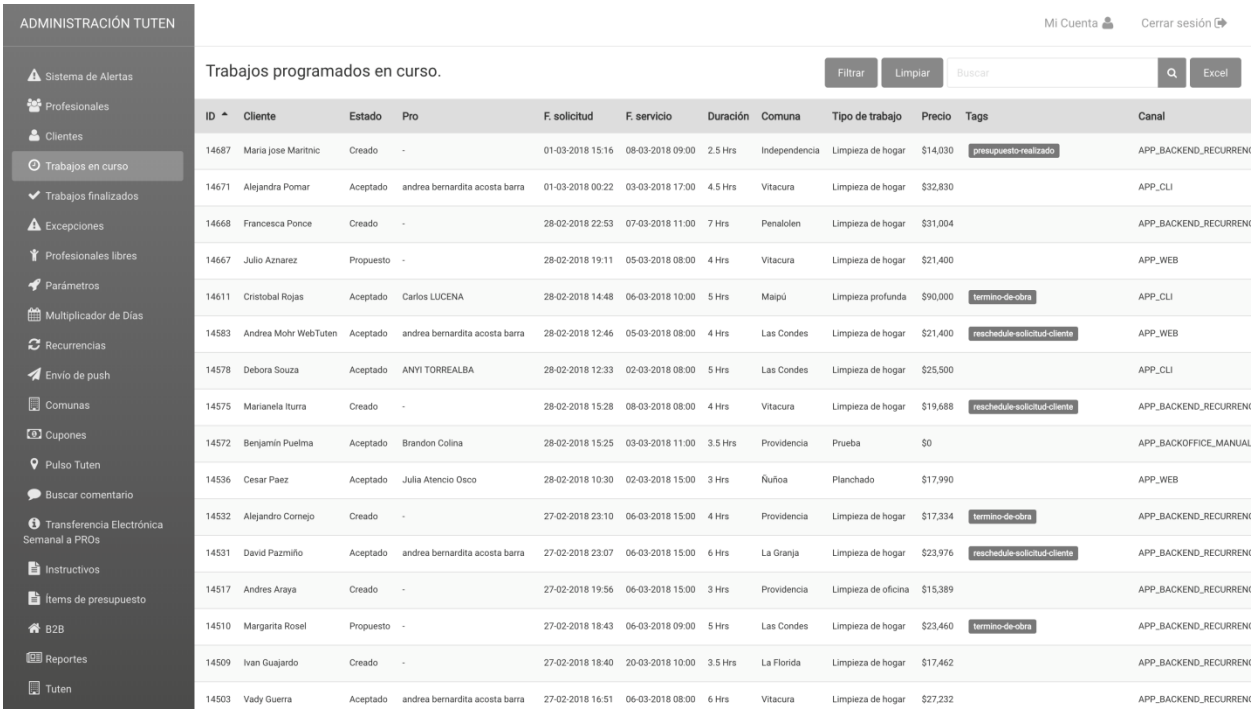
This platform allows the Tuten's operations area (or the company using the service) to carry out the basic operational functions to adequately provide *On-Demand* professional scheduling services.

It also presents a series of modules that allow the control of professionals, customers, and scheduled services, the administration of recurring services, and others. It also includes reporting modules, which provide support to management tasks.

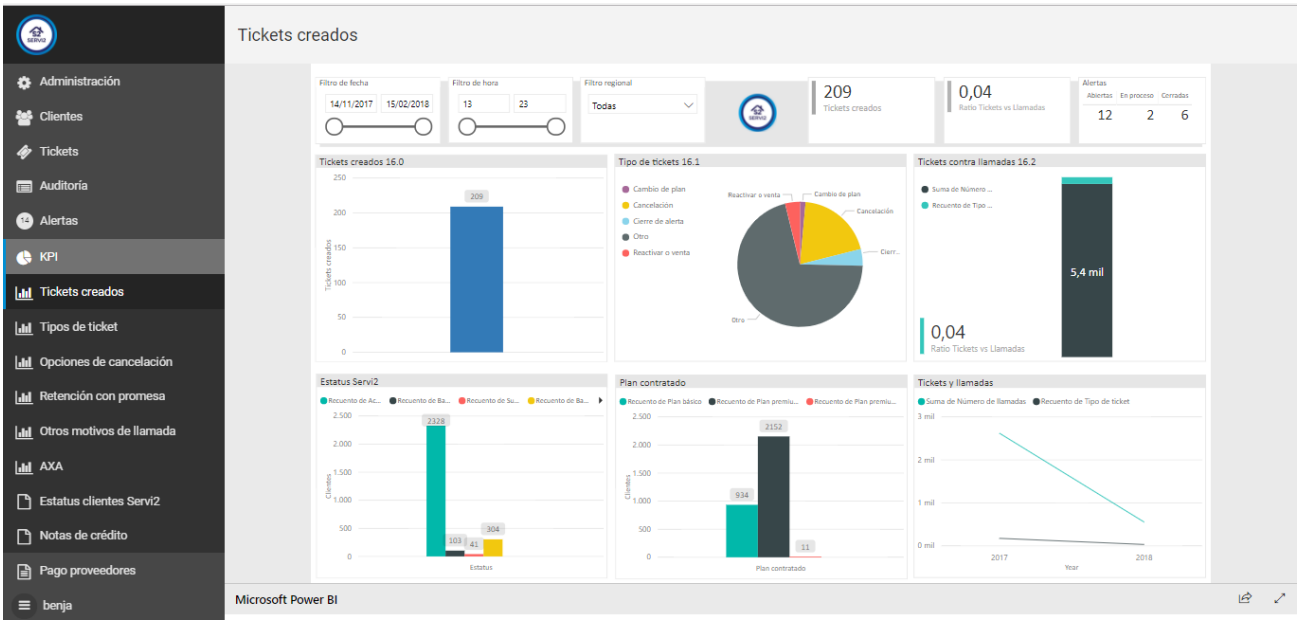
This module is widely used by the operations area to control the different services offered.

In addition to this, it presents several options that allow the management of services as well as the management of certain platform's configuration parameters (for example, activating/deactivating professionals, establishing parameters used by rules engines, etc.)

This website can be personalized in the event a company requests this service (B2B modalities), or request additional components (B2B SaaS modality).



In addition to the options mentioned above, Tuten offers *Business Intelligence* services, through which it is possible to generate different kinds of reports, thanks to the integration with Microsoft Power BI, based on each client's needs. These graphics are updated periodically and automatically, and are very helpful for the customers' business management and administration.



TUTEN CLOUD

This is the platform's core. All requests and jobs go through this Transactional Core hosted in a service cloud.

Sales, clients, professionals' databases, agendas, budgets, jobs in progress/scheduled/completed/rejected, and other elements of the platform are managed and controlled from here.

In this core, there is also an intelligent system that allows the *match* of services to be done, so that, based on business rules defined both for the task and by the company requesting the service, the best professional available for the job to be carried out can be located, using an efficient *Complex Event Processing* technology.

The architecture of this service cloud is explained in the following section, Architecture.

ARCHITECTURE

CLOUD SERVICES

It has been mentioned before that Tuten has a transactional core, where all transactions are managed. These services are provided by a network of servers and technological elements such as management systems for communication networks, and security and monitoring systems, among other elements. All services offered by Tuten will be located on this services cloud, as Tuten has decided that its entire transactional core shall be hosted in a services cloud.

A service cloud is a computational platform that allows the hosting of computational resources and assets, so they can be used with the purposes deemed convenient by the customer (in this case, Tuten), through a communications network (Internet). In our case, we use Microsoft's cloud infrastructure: Microsoft Azure.

To mention just a few:

- **Agility**
  For Tuten, it is essential to offer developments and *upgrades* in the shortest available time (we use agile methodologies as a project development system). In this context, the cloud provides agility and flexibility, as it allows new infrastructure to be available quickly in order to install and test new components. The same happens in the events of cloning systems that already operational.

- **Scalability + Elasticity**
  For Tuten, it is essential to offer a quality service in the shortest possible time. In the event that Tuten or a company requires a more computational infrastructure (for example, in the event of a specific situation where it is necessary to quickly have greater capacity), to scale the services and have a bigger capacity does not present a major difficulty, and can be managed in short and limited periods of time, given the way in which Tuten has built its business and infrastructure in the cloud.

- **High availability**
  In the occasional and uncommon case that certain servers become unavailable, Tuten has developed a high availability system, which allows requests to be re-routed to other servers offering the same functionalities in cases of error or unavailability of service. This are redundant systems used to minimize the impact of service unavailability, which are very uncommon (this will be explained in more detail in the next chapter)

- **Cost**
  As the services are hosted in the cloud, it is possible to decrease infrastructure, energy and equipment costs. The same happens by allowing the use of virtualization mechanisms, which allows us to scale economies.

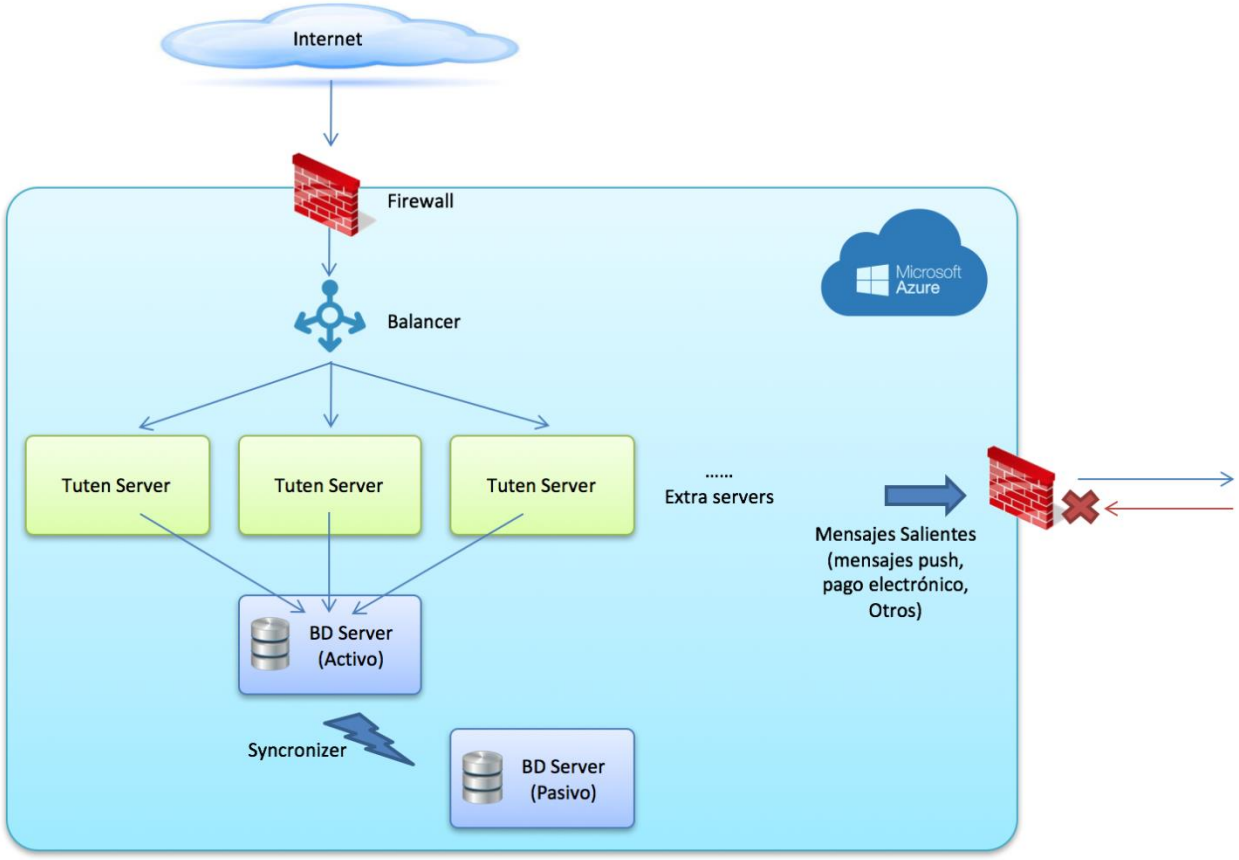
- **Performance**
  The fact that the systems are hosted in the cloud optimizes the use of computational resources in an automatic way, which allows a better platform monitoring, control and performance. This capacity provides transparency both for the customer and for the systems control in the organization.

- **Security**
  As the infrastructure is centralized, the cloud provider offers security systems added to its entire cloud, which are even better than those found in the traditional market. Although Tuten must control certain security aspects belonging to its platform (at the application level), the cloud offers security elements which support the Tuten business.

HIGH AVAILABILITY

Tuten has designed and implemented a technological platform that allows it to offer a high availability solution to its customers, which guarantees high levels of availability (SLA) in the market, in accordance with the requirements of a transactional network designed in a 7x24 modality.

The criteria used for the design adheres to demanding standards, with an operation in the cloud with redundant *sites* acting in an active-active modality in most of their layers, with the exception of the database layer, which operates in an active-passive modality due to this layer's definition and restriction, It also has a contingency model that allows the design to guarantee an RPO[1] and RTO[2] meeting our customers' requirements, based on a redundant infrastructure.



[1] Maximum point of data loss upon databases contingencies.

[2] Maximum time for the recovering of services upon platform contingencies.

SERVICES API

The system is designed as a web services architecture. For this, a services API has been defined, in which the services are used directly by Tuten with the aim of generating and establishing the system's complete operation. This API can be accessed via web service. It is of the REST type and it uses JSON as data exchange format.

All these services are self-documented for a better understanding of each of the services offered.

As a way of example, some of the services offered by this API are:

- admin:
  Administration operations

- booking:
  Operations related to the professionals' work

- funds:
  Operations related to system's funds (credits)

- location:
  Operations related to jobs, professionals and users georeferencing

- logs:
  Operations related to recording jobs, professionals and users activities

- params:
  Functions to operate the system's parameters

- payment:
  Payment operations for the system's jobs

- professional:
  Operations related to professionals and their characteristics

- user:
  Operations related to users and their characteristics

Although these services are defined in API modality, they are not of public access, nor can they be used by any system. There are business rules established that regulate what systems (internal or external) can use what systems.

TECHNOLOGIES USED

The platform has been developed using modern programming languages, a robust database engine, and other applications and systems supporting the technological and development tasks.

We briefly mention some of the technologies used below.

FRONTEND:

Tuten implements HTML5-based websites with JavaScript as programming language. CSS3 is used as style sheets at a centralized level, which allows the generation of visually attractive and normalized web pages.

To achieve this, Tuten relies on diverse *frameworks* and components, which allows the acceleration and optimization of the developments made in the frontend.

Just to mention a few, we have:

Node.js
It is a JavaScript execution environment, which allows for the creation of highly scalable network programs (in this case, web servers), as the functionality is not directly executed in the browser, but in the central server. Node.js is useful, as it provides a series of frontend functionalities.

This environment uses NPM as a package manager, which supports the use of Node.js, and is the one providing a range of functionalities necessary for Tuten's complete frontend.

AngularJS:
This *framework* allows an agile and orderly development, using structures of the MVC type (Model View-Controller) in HTML pages with CSS, and using JavaScript as event handler.

Coffescript:
It is a component responsible for compiling, interpreting and executing content on the controllers created with AngularJS.

Bootstrap:
It is a working *framework* that allows the acceleration of the coding and generation of sources, as it contains a series of components and libraries that allow us to develop web pages by using templates and snippets.

Grunt:
It is a task manager based on node.js libraries. It is used for the automation of development tasks necessary to optimize sources and to be able to execute JavaScript specific local tests.

Swagger:
It is a library used to establish a connection with the *backend*, between development environments, in a dynamic way. Swagger delivers information from the database to *frontend* components.

BACKEND:

Java
Java is a development environment, which allows the generation of both *backend* and *frontend* applications and components.

Tuten uses Java specifically to deliver micro-services and to support *backoffice* functionalities, among other services.

Glassfish
It is an application server, which contains the websites created for the *frontend*. It also provides all the services used by Tuten, using JEE as a service platform.

It also implements a REST services layer, for these to be used as a services API, both for Tuten and for companies that may use these functionalities.

Nginx
It is a lightweight and high performance application that allows the redirection of web requests between different servers. It allows both the balancing of requests and the performance of reverse proxy tasks and the control of certain access.

Tuten uses this application both to distribute the HTTP/HTTPS requests load, and to manage the access to the servers that must carry out the tasks.

DATABASES:

PostgreSQL
Tuten defined PostgreSQL as its database. This is a robust object-relational and Open Source RDBMS, highly customizable, with concurrent management, low latency levels, and a high degree of utilization by the community (which guarantees wide support and knowledge in the event of bugs).

Additionally, this database allows the implementation of replication, redundancy and high scalability services, while it offers all the services of a relational database.

MOBILE TECHNOLOGY:

Ionic

This is a hybrid mobile application development *framework*, which allows us to compile apps to work in Android, iOS and Windows Phones, using JavaScript as programming language.

OTHERS:

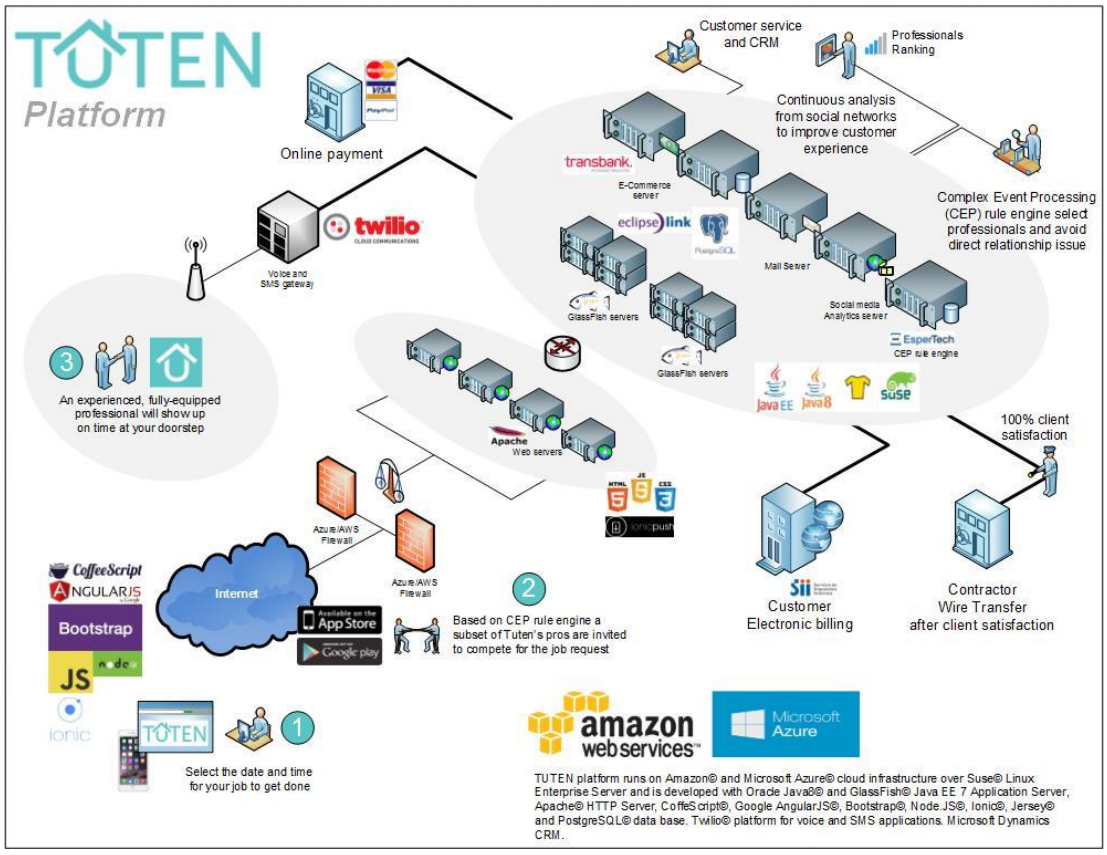
Mail Server:
Google Mail is used as email infrastructure.

Push Notifications
Firebase Cloud Messaging (FCM) is used as a system that allows push notifications on the Customers' and Professional App.

IVR
The Twilio API is used for sending voice messages (IVR) and SMS to customers.

Webpay
For electronic payments, the webpay plus and oneclick API is used. For this, a proprietary payment *gateway* was built, which allows the establishment of communication mechanisms between these systems, with security at both ends.

An isometric diagram including the different components, technologies and they way in which they interact is presented below.



SECURITY

Tuten offers different and varied security components, allowing both the operation and the access to information by means of appropriate security mechanisms.

In the first instance, the site tuten.cl (where all the applications are born) is hosted in a Microsoft Azure services cloud, guaranteeing an extraordinary resistance to attacks on the Azure environment. There exists continuous supervision and controlled access over the entire cloud. These service clouds guarantee that the access to the information meets international standards, where even the access to the racks where servers are contained (in the cloud datacenter) is controlled, authorized and audited, to prevent undesired fraud and physical hacking.

This site uses HTTPS as communications protocol, which guarantees the encryption of communication. TLS 1.2 is used as encryption protocol. The site has a digital certificate issued by trusted entities recognized worldwide, and renewed every so often to ensure reliability.

It is necessary to mention that the access to the database always takes place from known and established systems. Access is controlled to guarantee that only certain pre-established applications can use it. The information contained in the database cannot be accessed by other systems. The same happens with systems foreign to the organization. In this sense, we make sure to keep the database systems isolated, as an extra security measure within the information control.

Each of the services using of database information is properly controlled with firewall and proxy rules, by means of users and authentication credentials.

All access to the database is controlled and duly authorized.

DEVELOPMENT METHODOLOGIES

SCRUM

Together with the quality of delivered products, at Tuten, we strongly believe in agility and collaboration. In this sense, as a company, we have decided to implement each development made using agile methodologies. After evaluating and testing several alternatives, Scrum was established to be used as a software development methodology.

Scrum proposes a series of premises which, as a whole, allow each functionality to be grouped into relatively small (called *sprints*) *releases* (deliverables), and therefore, to be validated by the client in limited and not long periods (*time-to-market*). This is essential, as it allows applicants to see products regularly, and with this, to determine whether modifications that may not have been initially not contemplated and/or new applications are needed in very short periods. Additionally, it allows the reduction of non-compliance risk, reducing costs (as it prevents the increase of project gaps and total project time), among others.

An interesting feature relates to risk decrease. By issuing deliverables relatively quickly, the risk related to, on the one hand, the project not meeting the client's expectations is mitigated, preventing non-compliance with deadlines due to undesired developments (where the client realizes the deliverable is not what he/she really needed and/or requests other non-requested functionalities), decreasing the number of gaps. It also helps the early correction of unwanted functionalities, which reduces the amount of future gaps, as well as the final cost of the project, controlling both time and cost deviations in a timely manner.

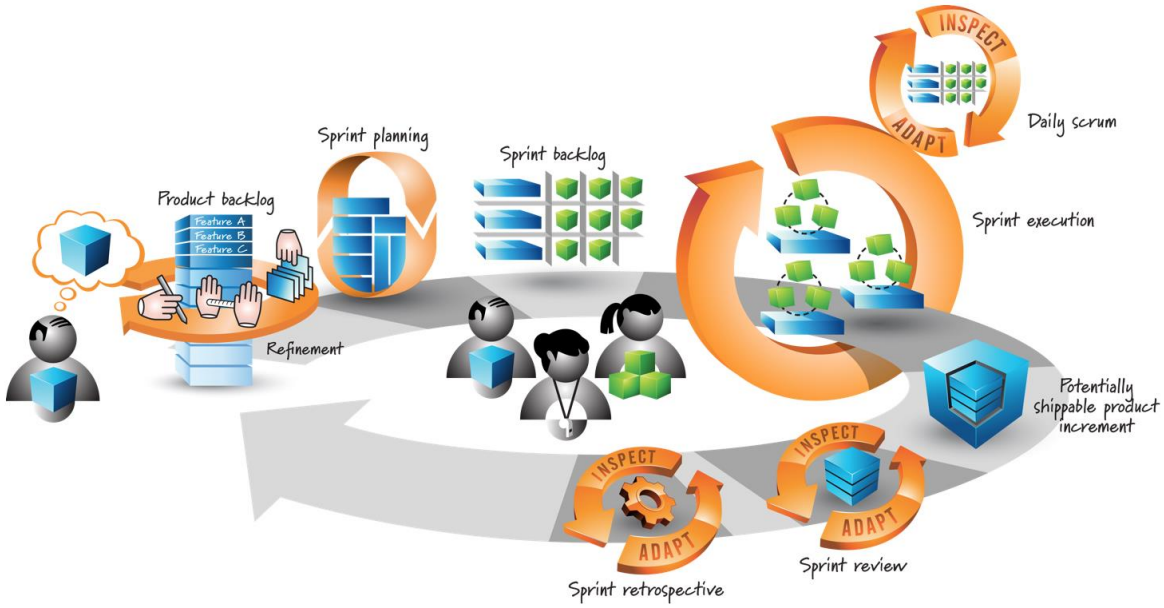
By themselves, these qualities constantly motivate the development teams, as the teams can see early successes and the fruition of their projects.

Simple stated, the main benefits are:

- Flexibility to changes
- *Time-to-market* reduction
- Higher software quality
- Productive improvements
- Risk reduction
- Transparency and traceability
- Future predictions of time for upcoming deliverables

The methodology also has a very attractive feature for Project Managers and customers: Developments can be carried out both in an evolutionary (the next *release* is a series of new features annexed to the previous development) and iterative way (the next *release* are more complete functionalities in comparison with the previous *release*). This flexibility is very useful when planning how to perform the requested development with the client.

To achieve the aforementioned characteristics, it is necessary to meet a series of elements that are typical of this methodology. In the following diagram, a graphic showing how this methodology works appear. Below it, we will explain briefly how Tuten has implemented the use of this methodology.



1. Establishment of requirements (Backlog definition)
   Tuten has defined that the process of taking requirements is key for the project's success. In this way, both the development team and the Product Manager, together with the client, participate in establishing the requirements. Each of the requirements is written in Stories, which are documented in an Establishment of requirements file (Product Backlog).

   While the process of establishing requirements takes place, each story is refined until sufficient detail is obtained regarding the client requirements, and documented for it to be used by the developers.

2. Planning (Sprint definition)
   With the defined *Product Backlog*, each of the tasks to be performed is assigned, based on the development team's know-how, as prior metrics. Here, deliverables are prioritized and grouped, defining the number of deliverables to be made (how many *sprints* will it take to develop the entire product). This process is called *Sprint Planning*, and is developed jointly by the Product Manager and the development team.
   With the definition of each sprint and each task within the sprint (sprint backlog), the customer is consulted on whether the delivered planning matches his/her needs. In case the modification of some tasks, the addition of new tasks, or the modification of the priorities of some of them is needed, this process is done once more, until it is agreed upon by all the parties involved.

3. Collaborative work
   Once the Sprint Backlog is defined, the development team begins the coding tasks (*Sprint Execution*). Each developer undertakes some of the assigned tasks and begins the development established in the *story*.

   While the execution of the Sprint takes place, regular meetings must be held to monitor the project's progress .

4. Follow-up meetings
   Scrum defines several types of follow-up meetings. Tuten uses mainly two of them:

   - *Daily Scrum Meeting*
     On a daily basis, the development team meets, together with the Product Manager, and reviews the project's progress. This should ideally be done at the same time daily.
     This meeting is very express and concise. Each developer is expected to explain how the development of the entrusted tasks is going, in a time of no more than 5 minutes. Only those involved in the sprint can speak.

     The developer must answer 3 questions:
     - What did he/she do the day before?
     - What will he/she do today?
     - Explain the problems he/she has encountered during the development of the task (to check how this impacts on the sprint's development, and to evaluate alternatives in order to avoid deviations in the course of the sprint)

   - *Weekly Scrum Meeting*
     This are meetings where the Product Manager provides the clients with information regarding the progress and the compliance degree.

5. Internal and customer review (QA and UAT)
   Once all the tasks associated with the Sprint Backlog have been completed, the Product Manager, together with the QA (Quality Assurance) area, reviews each of the Sprint's tasks, looking for potential errors and/or non-fulfillment of what has been requested (Sprint Review). In case modifications are needed, they are re-directed to the development team for them to apply them.

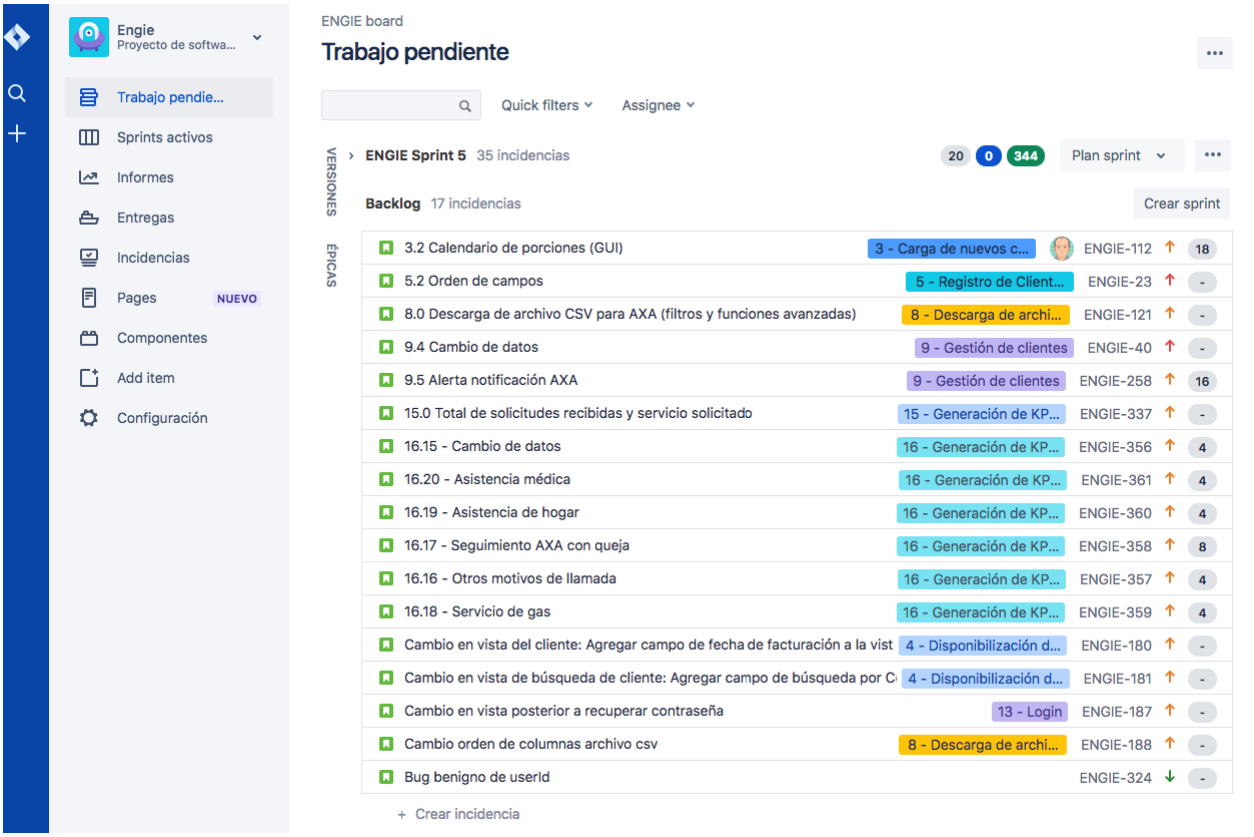
   Whenever the quality of the deliverable is appropriate, all the components are delivered to the UAT environment, in order for them to be tested by the customer.

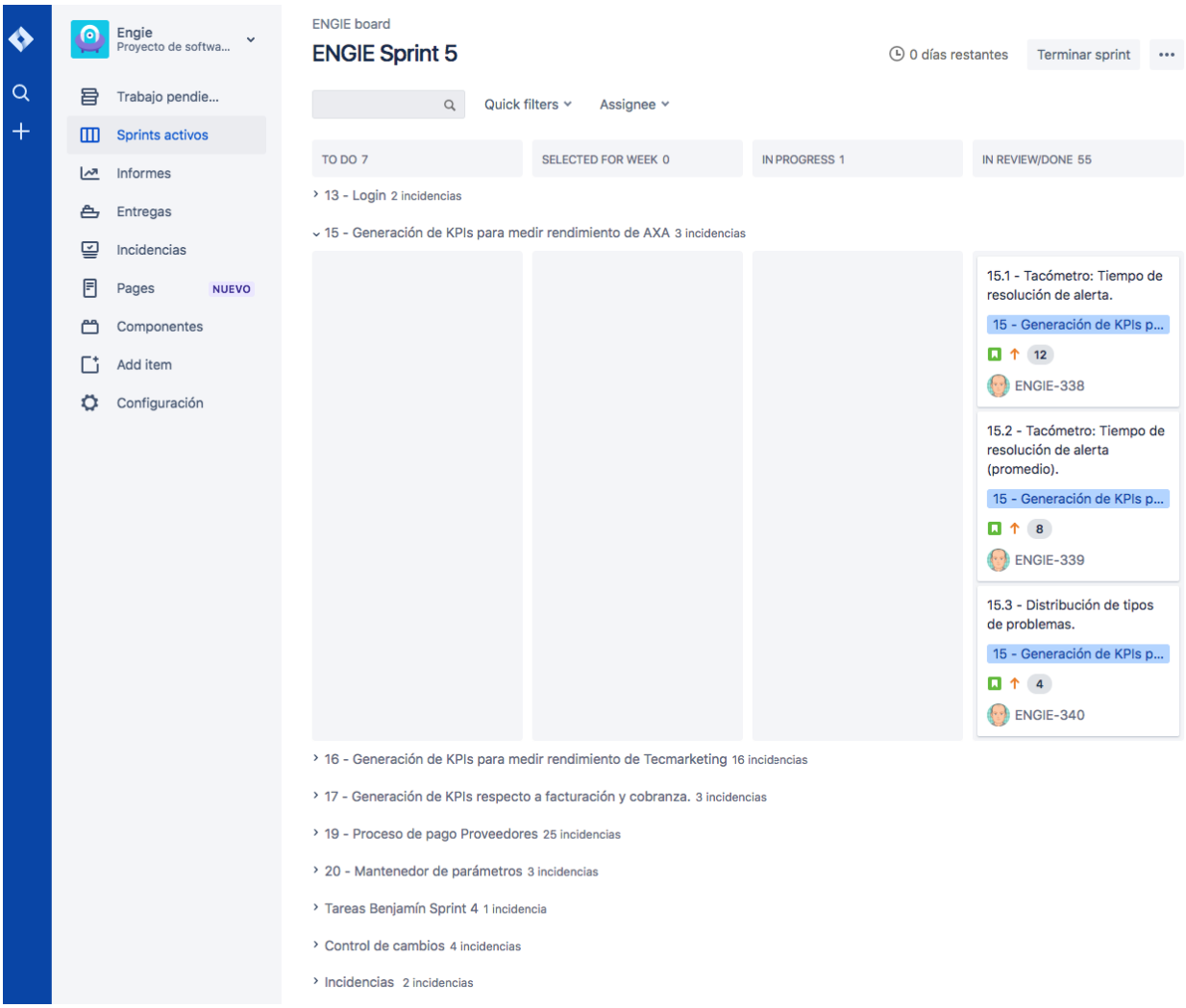
6. Retrospective Review

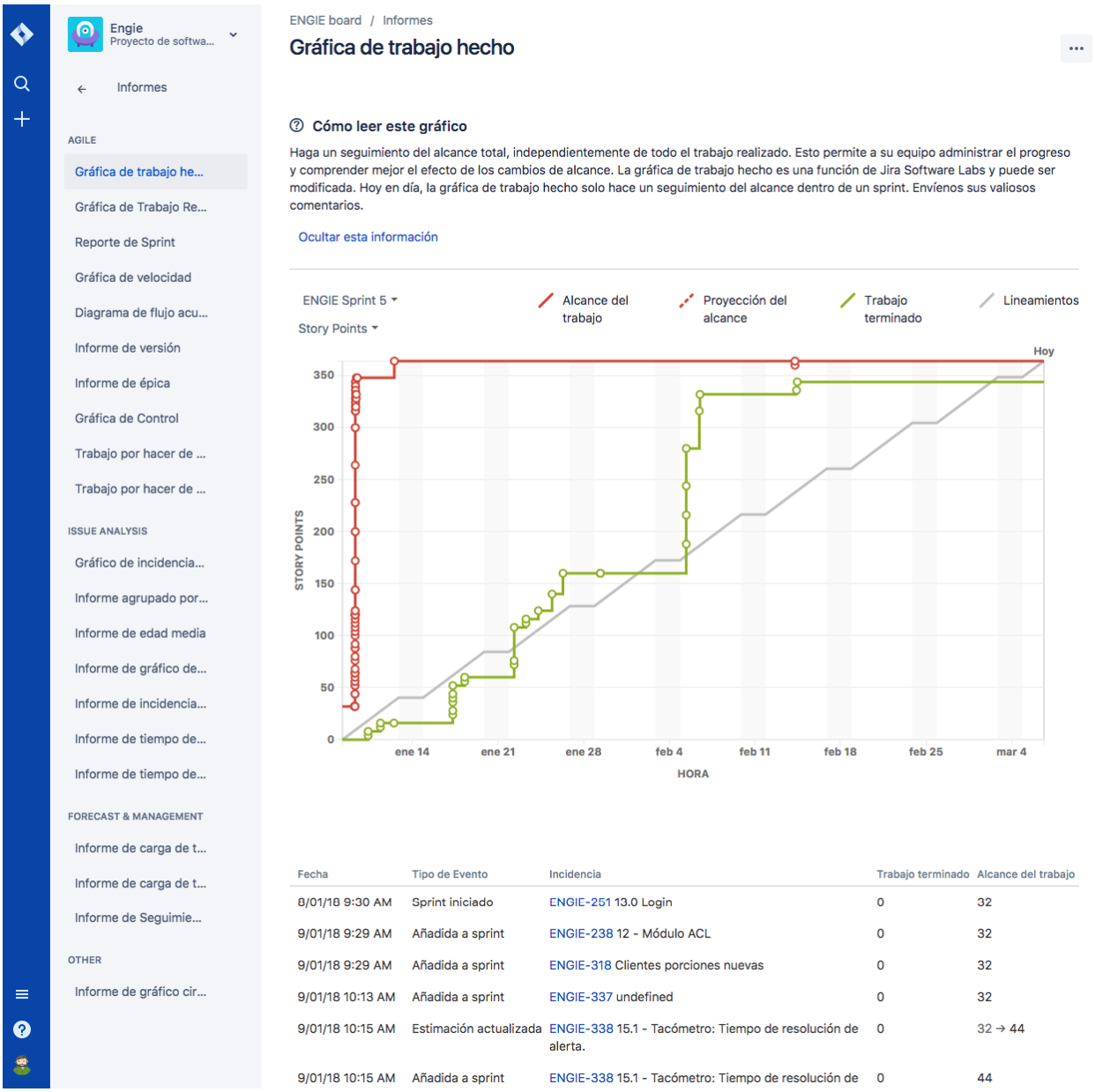
Once the sprint is completed, the project team meets to analyze the impressions of the recently completed sprint. The purpose of this review is to determine future points of improvement (continuous improvement), with the aim of defining lessons for the next Sprints. Additionally, modifications of the applied times must be documented (particularly deviations of the initially estimated deadline), in order to be able to establish more precise future estimates.

As a way to support the development management, several tools are used, such as Skype, Zoom, Slack, Balsamiq Mockups, version control systems, Atlassian Jira, and others.

Regarding Jira, Tuten uses this platform to control each of the sprint's and the projects in general tasks. This website is especially effective for controlling and monitoring of each task.




Engie
Proyecto de softwa...
Trabajo pendie...
Sprints activos
Informes
Entregas
Incidencias
Pages
NUEVO
Componentes
Add item
Configuración
ENGIE board
ENGIE Sprint 5
0 días restantes
Terminar sprint
Quick filters
Assignee
TO DO 7
SELECTED FOR WEEK 0
IN PROGRESS 1
IN REVIEW/DONE 55
13 - Login 2 incidencias
15 - Generación de KPIs para medir rendimiento de AXA 3 incidencias
15.1 - Tacómetro: Tiempo de resolución de alerta.
15 - Generación de KPIs p...
12
ENGIE-338
15.2 - Tacómetro: Tiempo de resolución de alerta (promedio).
15 - Generación de KPIs p...
8
ENGIE-339
15.3 - Distribución de tipos de problemas.
15 - Generación de KPIs p...
4
ENGIE-340
16 - Generación de KPIs para medir rendimiento de Tecmarketing 16 incidencias
17 - Generación de KPIs respecto a facturación y cobranza. 3 incidencias
19 - Proceso de pago Proveedores 25 incidencias
20 - Mantenedor de parámetros 3 incidencias
Tareas Benjamín Sprint 4 1 incidencia
Control de cambios 4 incidencias
Incidencias 2 incidencias



Fecha	Tipo de Evento	Incidencia	Trabajo terminado	Alcance del trabajo
8/01/18 9:30 AM	Sprint iniciado	ENGIE-251 13.0 Login	0	32
9/01/18 9:29 AM	Añadida a sprint	ENGIE-238 12 - Módulo ACL	0	32
9/01/18 9:29 AM	Añadida a sprint	ENGIE-318 Clientes porciones nuevas	0	32
9/01/18 10:13 AM	Añadida a sprint	ENGIE-337 undefined	0	32
9/01/18 10:15 AM	Estimación actualizada	ENGIE-338 15.1 - Tacómetro: Tiempo de resolución de alerta.	0	32 → 44
9/01/18 10:15 AM	Añadida a sprint	ENGIE-338 15.1 - Tacómetro: Tiempo de resolución de	0	44

CONTINUOUS INTEGRATION

Tuten has defined agility as a key element and as a differentiator in all its software development processes. For the software delivery process to be really effective, it requires high efficiency levels in the whole process, strong communication and collaboration between the work teams, and the technological infrastructure to be able to establish differentiating elements contributing value to the agile management of deliverables. In simple terms, an agile work culture is required.

This required the adoption of a software engineering practice called DevOps (an acronym of the terms development (software development) and operations (IT operations)). This practice defines some technological elements that allow software development tasks to be supported, in terms of automation and monitoring of IT infrastructure, which supports software construction processes, through testing, implementation in production, and software administration processes of the IT infrastructure.

One of the most important elements relates to continuous integration processes. This allows us to have very stable systems in productive environments, which guarantees the operational continuity.

Some elements allowing the support of the continuous integration processes are:

1- Version control.
   Tuten uses a version control system in the cloud (Bitbucket).
   This repository allows us to have different versions of the System and to define execution environments (development, certification, production).

2- Cloud orchestration system (Codeship).
   This system takes each version of Tuten and carries out the Software construction and its automatic tests. In the cases when the construction and tests do not present, and depending on the environment, the new version will be installed in the corresponding environment.

3- Automatic Testing System.
   JUnit is used as an automated unit test management system.

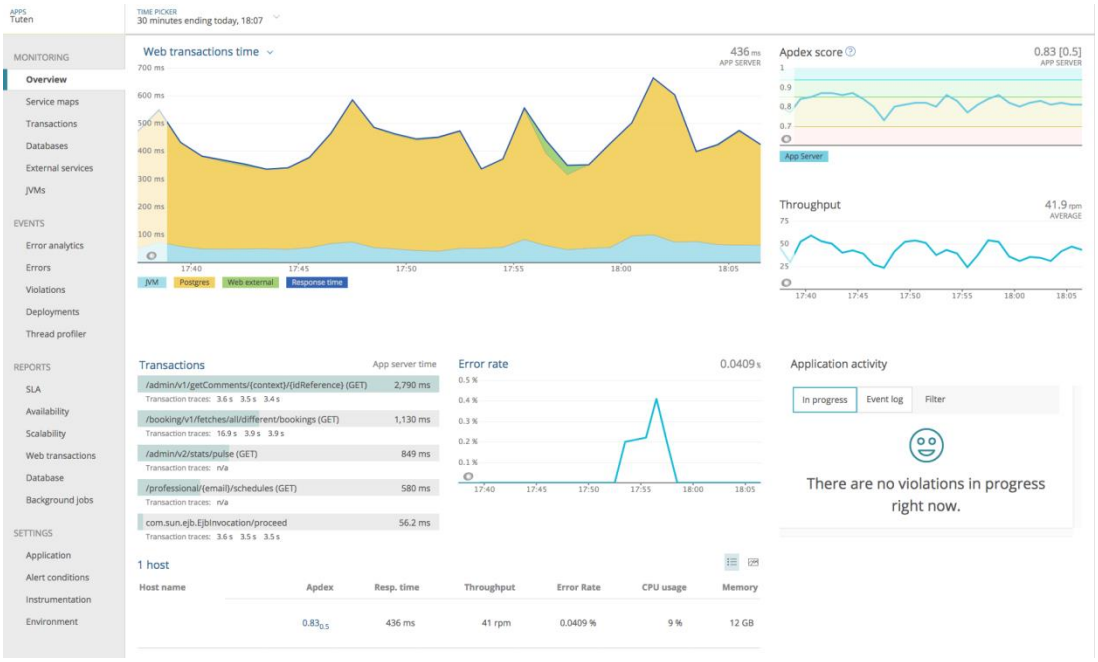
4- Automatic construction system.
   Tuten uses Maven 2, as it allows us to build Software (compile) to be used, by means of a construction description file.

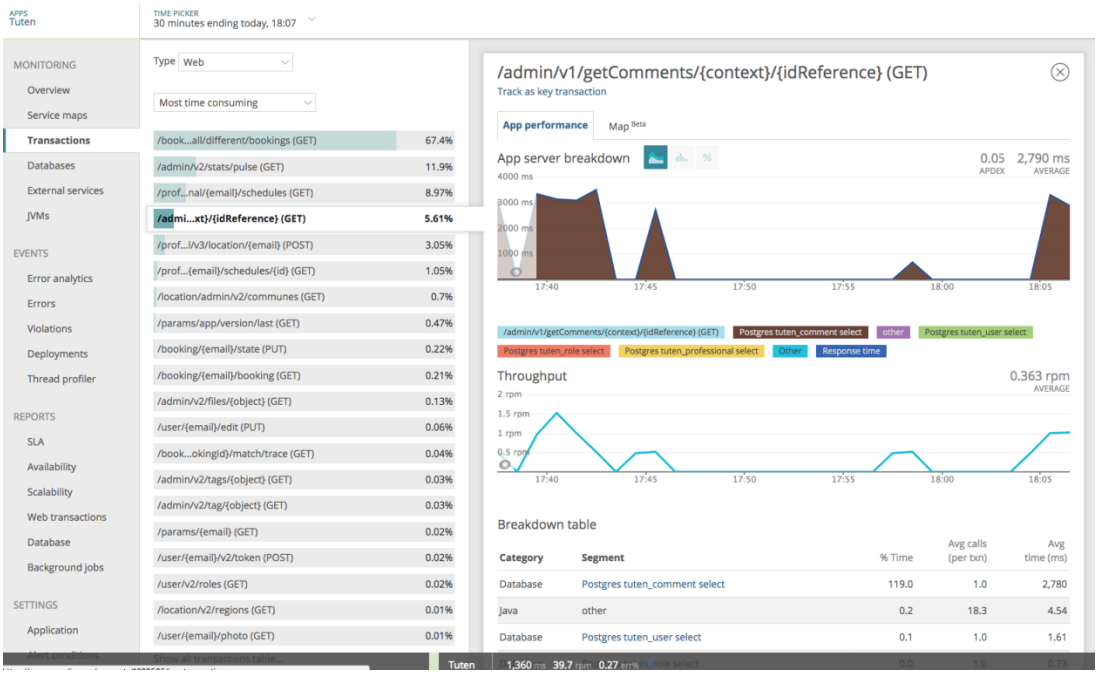
5- Automated releases.
   Tuten uses systems that automate the releasing deliverables, executing unit tests, and scheduling tasks processes. For these automation tasks, Jenkins and Rundeck are used.

6- Monitoring.

We also count on some monitoring tools that allow us to review infrastructure behavior in real time, and/or within time ranges. This is especially useful in productive environments, where it is necessary to control and monitor the platform's performance at all times.

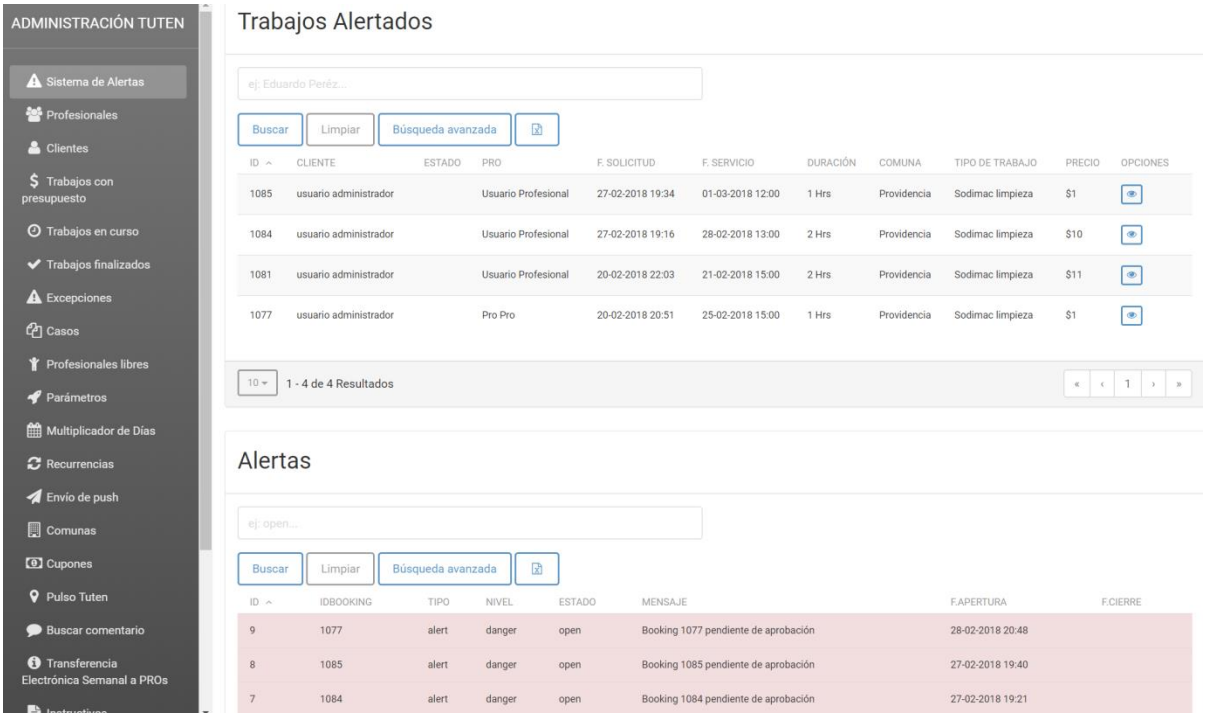
To support this, different elements are used, one of them being New Relic.

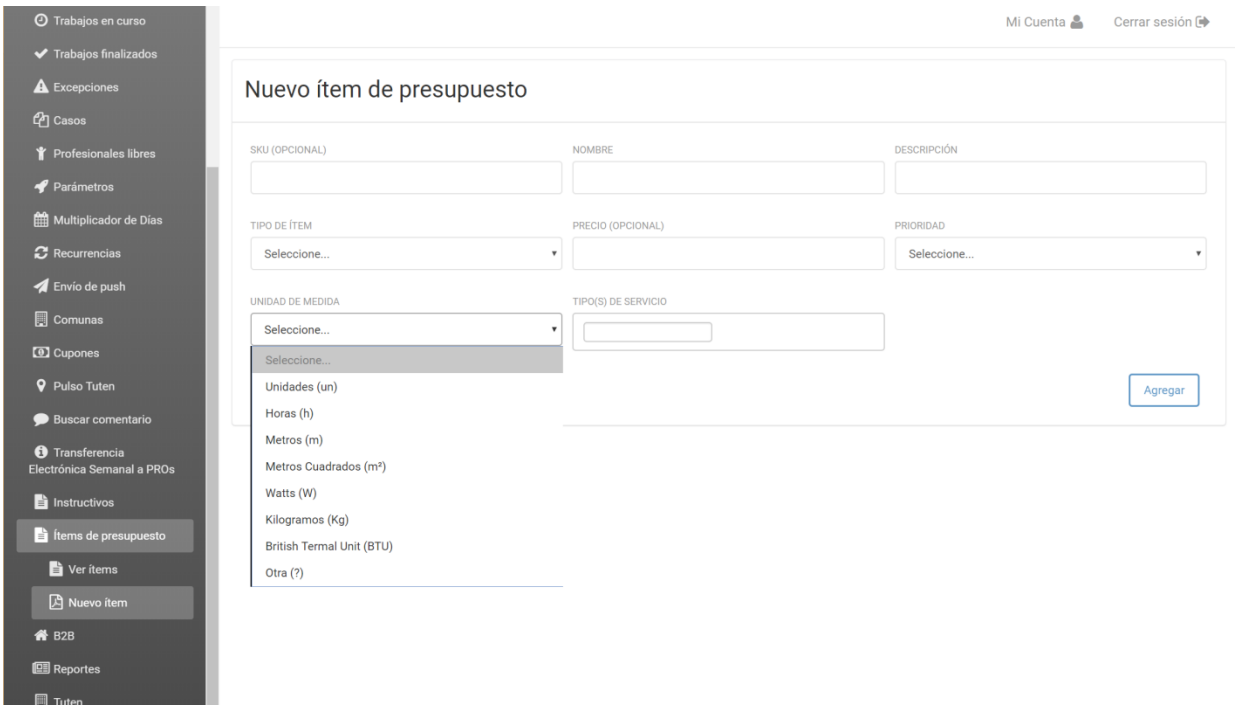


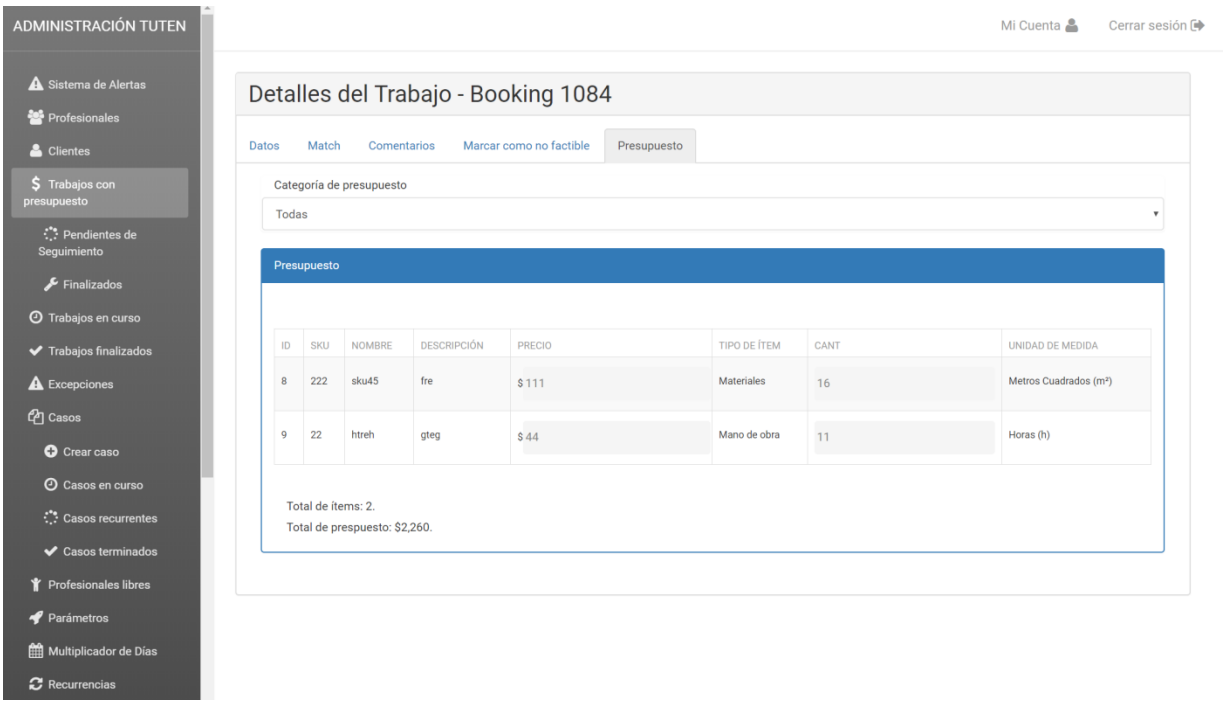


ANNEXES

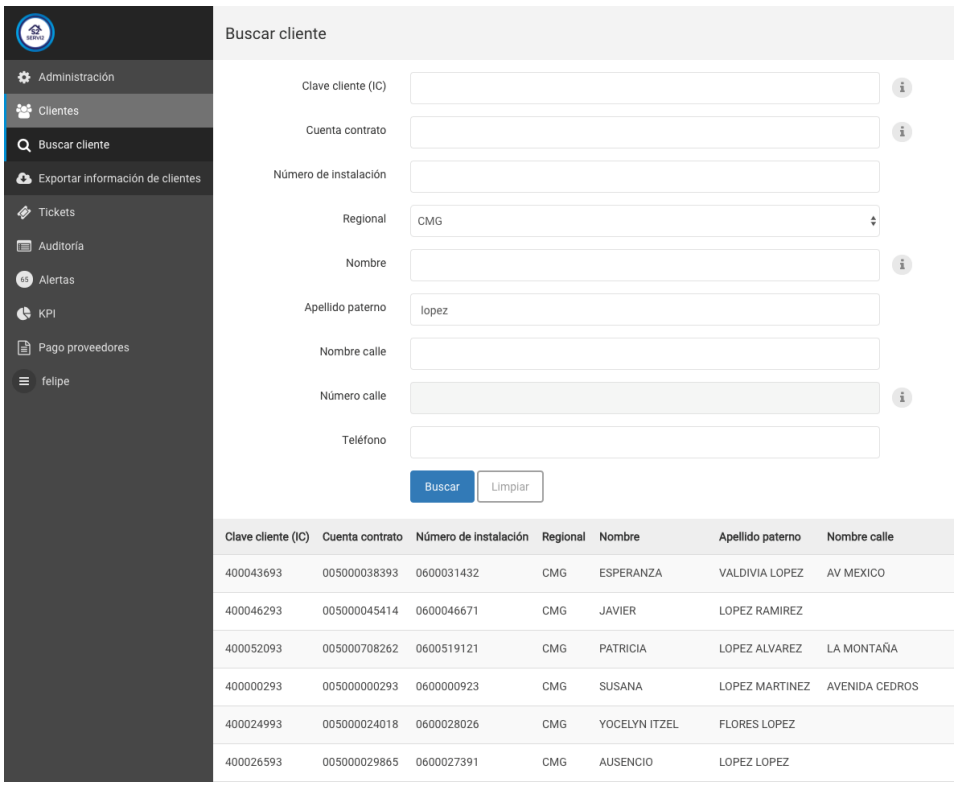
ANNEX 1: EXAMPLES OF ALERTS AND BUDGETING FUNCTIONS IN TUTEN

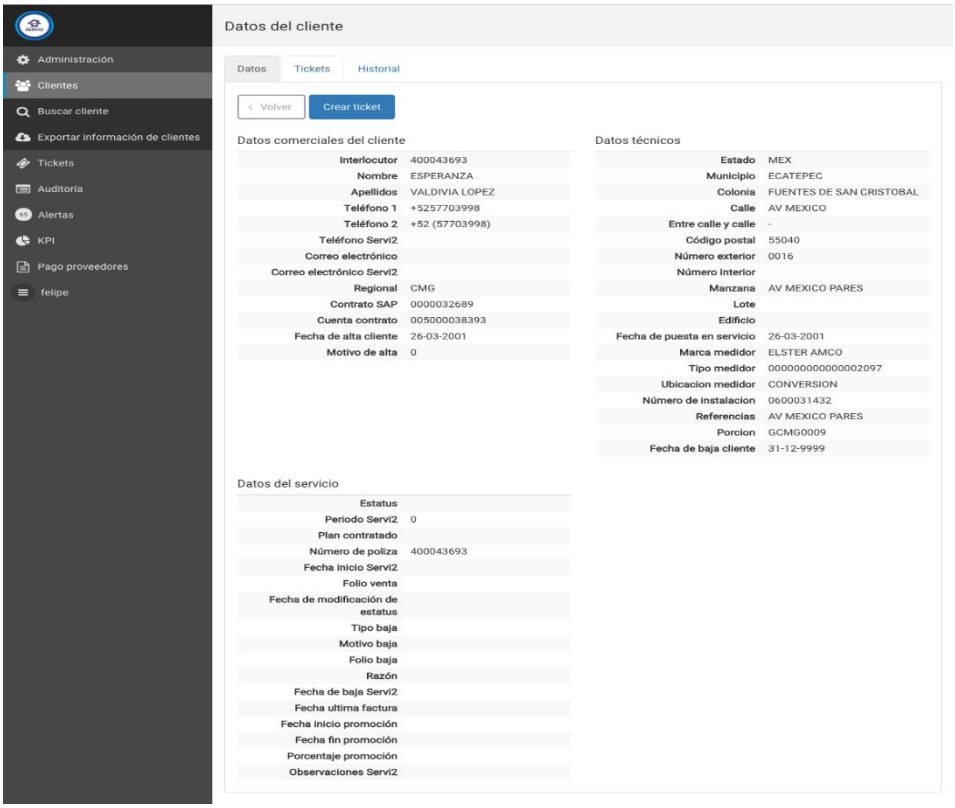


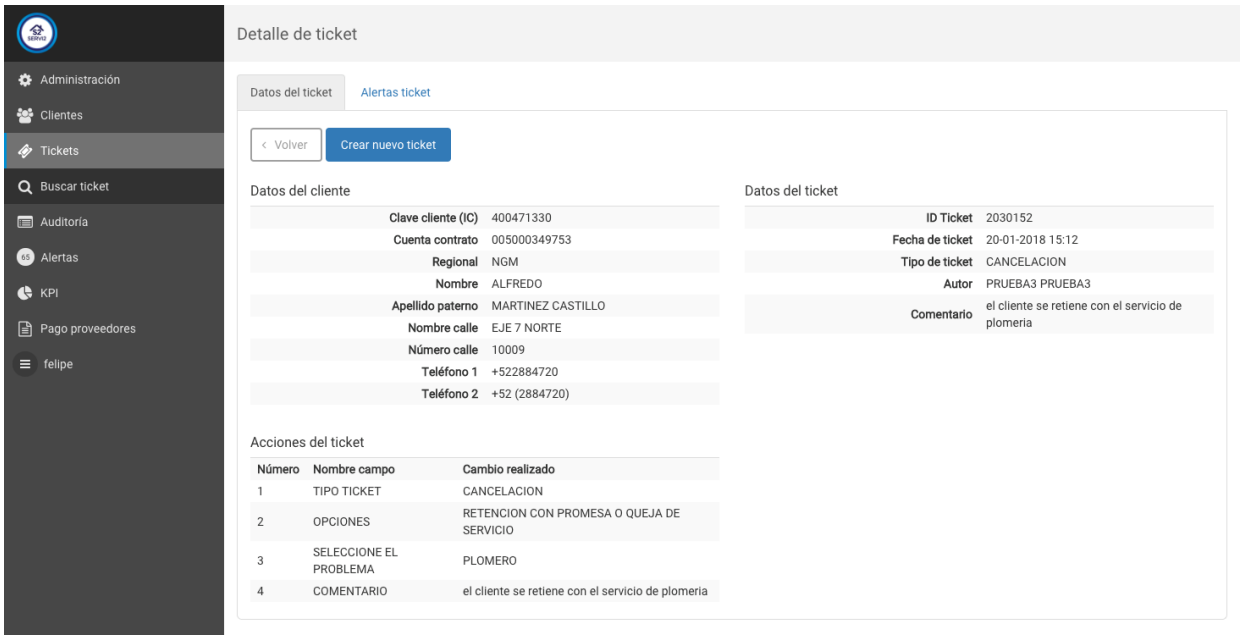


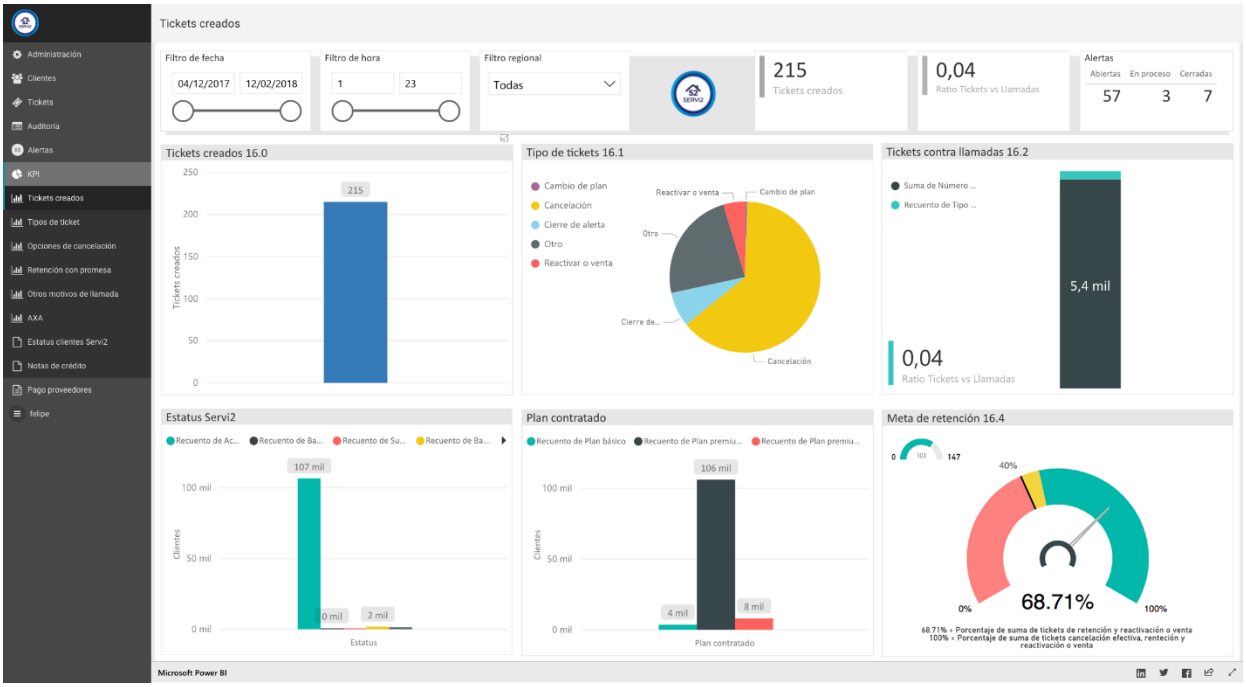

ADMINISTRACIÓN TUTEN
Sistema de Alertas
Profesionales
Clientes
Trabajos con presupuesto
Pendientes de Seguimiento
Finalizados
Trabajos en curso
Trabajos finalizados
Excepciones
Casos
Crear caso
Casos en curso
Casos recurrentes
Casos terminados
Profesionales libres
Parámetros
Multiplicador de Días
Recurrencias
Mi Cuenta
Cerrar sesión
Detalles del Trabajo - Booking 1084
Datos
Match
Comentarios
Marcar como no factible
Presupuesto
Categoría de presupuesto
Todas
Presupuesto
ID
SKU
NOMBRE
DESCRIPCIÓN
PRECIO
TIPO DE ÍTEM
CANT
UNIDAD DE MEDIDA
8
222
sku45
fre
$ 111
Materiales
16
Metros Cuadrados (m²)
9
22
htreh
gteg
$ 44
Mano de obra
11
Horas (h)
Total de ítems: 2.
Total de prespuesto: $2,260.

ANNEX 2: EXAMPLE OF B2B SAAS ENGIE SERVI2 PLATFORM






Detalle de ticket
Administración
Clientes
Tickets
Buscar ticket
Auditoría
Alertas
KPI
Pago proveedores
felipe
Datos del ticket
Alertas ticket
Volver
Crear nuevo ticket
Datos del cliente
Clave cliente (IC) 400471330
Cuenta contrato 005000349753
Regional NGM
Nombre ALFREDO
Apellido paterno MARTINEZ CASTILLO
Nombre calle EJE 7 NORTE
Número calle 10009
Teléfono 1 +522884720
Teléfono 2 +52 (2884720)
Datos del ticket
ID Ticket 2030152
Fecha de ticket 20-01-2018 15:12
Tipo de ticket CANCELACION
Autor PRUEBA3 PRUEBA3
Comentario el cliente se retiene con el servicio de plomeria
Acciones del ticket
Número Nombre campo Cambio realizado
1 TIPO TICKET CANCELACION
2 OPCIONES RETENCION CON PROMESA O QUEJA DE SERVICIO
3 SELECCIONE EL PROBLEMA PLOMERO
4 COMENTARIO el cliente se retiene con el servicio de plomeria


Tickets creados
Filtro de fecha
04/12/2017
12/02/2018
Filtro de hora
1
23
Filtro regional
Todas
215
Tickets creados
0,04
Ratio Tickets vs Llamadas
Alertas
Abiertas En proceso Cerradas
57 3 7
Tickets creados 16.0
Tipo de tickets 16.1
Tickets contra llamadas 16.2
Cambio de plan
Cancelación
Cierre de alerta
Otro
Reactivar o venta
5,4 mil
Estatus Servi2
Plan contratado
Meta de retención 16.4
68.71%
Microsoft Power BI

ANNEX 3: EXAMPLE OF B2B SAAS PLATFORM FOR A/C INSTALLATION