

Smart Health: Mobile Application for Booking Medical Appointments

Liliana Enciso^{1,2}, Jonathan Castro² and Elmer Zelaya-Policarpo²

¹*Grupo Investigación IoT - Network Communications on Smart Environment, Universidad Técnica Particular de Loja, San Cayetano Alto, Loja, Ecuador*

²*Departamento de Ciencias de la Computación y Electrónica, Universidad Técnica Particular de Loja, San Cayetano Alto, Loja, Ecuador*

Keywords: Calendar, Reserves, Webservice, Smart Health.

Abstract: The article describes the development and implementation of a web application developed in python through django framework, whose main functionality is the reservation of appointments and the sending of them via e-mail and text messaging to the recipients. The operational model indicates the infrastructure and the process of use. In this work the device is oriented towards the field doctor, where work schedules are pre-established and the interface is developed properly for doctors or professions related to medicine without previous or advanced knowledge in computing. The frame views and the final interfaces of this application are also presented.

1 INTRODUCTION

At present, we live an autonomous technology of renewal of wireless technologies and automatic communication systems, with the passing of years thankfully for researching and technological development (Gralla and Lindley, 2006).

The development of a web application for records of clinical reserves, hospitals, private practices, solve the problem of ordering appointments or forgetting them, both at the personal level of the doctor and or receptionist, as well as the patient (Hassan et al., 2004).

This way of storing patients clinical reserves in a hospital limits their attention because for different reasons a person can change the date or time of care, thus, initiating in the new establishment another clinical reserve, hindering their continuity in care, due to The traditional systems that are used, in diaries, books, that require a greater search time than an automated system with a patient search engine (Hassan et al., 2004). Likewise there are other applications in the health context that allow the geolocation of medical centers (Enciso-Quispe et al., 2018b) and others customize medical alerts using DHIS2 tools (Enciso-Quispe et al., 2018c).

In the case of the doctors and profesional people Loja city, the archives represent an analogous case where huge amounts of information are ordered, specifically in the case of appointments, it is clear how

effective, functional and practical the databases can be in the administration of the reserves and the state in which they are (Thompson and Antezana, 2015).

There are currently low reserve systems of annual subscription of approximately 500 US dollars with email sending, which does not allow to know the status of the appointment, it is important to mention that it also does not allow to store a database of patients, these types of tools do not allow reminder to the Cell phones, because there is no global mobile operator in Ecuador. Also they are confusing tools without defined schedules that turn the process, slow for de user (Enríquez et al., 2007).

2 RELATED WORKS

Google calendar.- It is able to synchronize a mail tray to place reservations confirmations, can create, edit and delete events, send email and send text message (McDonald et al., 2011).

Jorte Calendar.- This application is functional on all mobile platforms, this organizes a real role with an attractive and easy to understand appearance, has the function and the possibility of adding photos to events.

It has a very special design with a great emphasis on the color combinations and icons of the tasks, since you know what you have 30 days a month, I can send text messages and send mail. However, it does not

fit into the organizational model of a professional or health entity (Álvarez-García et al., 2011).

Today calendar.- It allows to visualize the events of an easy, fast and understandable form for the user, this calendar owns items for reservation of appointments this also allows only the sending of appointments by email way.

Business calendar 2.- It is one of the most complex applications to organize daily diary, it is a calendar designed specifically for business people, it has been designed and improved for functionality, efficiency and interface, through features for administrative functions, this application allows sending e-mail And text messages. Previous acknowledgement in computing is recommended because of its complexity.

Sol calendar.- It is one of the complete calendars has an interface with 3 views available per month, per day and year, shows the events by time slots distributed throughout the hours, while the calendar shows the events in the form of a list of tasks, It also allows sending appointments via text message and email.

All the applications developed are of an international nature, there is no local proposal for appointment booking that works through the web, in Ecuador the only system of sending text messaging is from banks, with private, non-commercial systems, That is to say, they are not available to people whose offices are small or people who need to make reservations whose technological infrastructure is limited (Arenburg et al., 2010).

3 THEORETICAL FRAMEWORK

3.1 Web Applications

Until a few years ago the Web was just collections of pages, documents and static and simple images that people could consult but actually its possible to interact with them. An important step in the evolution of the Web was the creation of CGI (Common Gateway Interface) which defines a mechanism where information can be passed between servers and certain external programs (Erl, 2004).

The solution to this comes basically in 2 ways:

A. Module execution systems are designed that are better integrated with the server, avoiding the installation and execution of several programs.

B. The servers are given to an interpreter of some type of programming language that allows to include the code in the pages in a way that the server executes it.

3.2 SCRUM Methodology

Scrum is a process in which a set of good practices are applied on a regular basis to work collaboratively, as a team (Enciso-Quispe et al., 2018a), and to obtain the best possible result of a project. These practices support each other and their selection originates in a study of the way of working of highly productive teams (Martel, 2014).

3.3 Data Base

From the computer point of view, the Database is a system consisting of a set of data stored on disks that allow direct access to them and a set of programs that manipulate that data (Hammer and McLeod, 1978).

Each database is composed of one or more tables that hold a dataset. Each table has one or more columns and rows, where each column stores a portion of the information about the appointments, the hours, the query reason, the status of the appointment (Douglas and Douglas, 2003).

3.4 Mobile Services

Nowadays, due to the great boom of mobile services based on text messages, there are a large number of reliable and useful solutions with remarkable features; Necessary to send text messages manually or automatically from a computer / server to provide notification services, news, etc (Gallardo et al., 2007).

SMS is connected to the cellular network with a set of services or SMS applications, for which it usually provides a series of interfaces so that software developers can receive, process and send a high number of messages. The systems that link an SMS gateway can be seen better (Harb et al., 2008).

Through an Android API its possible to send text messages, however, to retrieve data from the web to the cellphone can do with queries in PHP through a data retrieval in a JSON that will be broken down and placed in fixes, For the sequential sending of messages, as well as scheduling the exceptions in case there are incorrect numbers or lack of mobile balance in the pone (Delegido, 2016).

3.5 Architecture

The reservation application is developed in the django framework that allows programming with the python programming language, it has by default an architectural model which we will use as the model, view, controller (Ross et al., 2006).

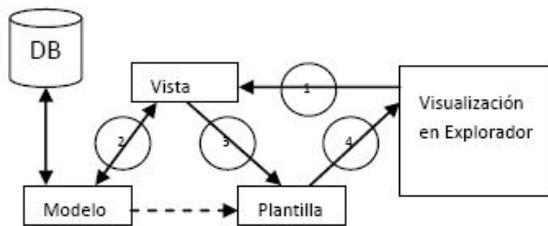


Figure 1: MVC Django Framework.

The model that uses the django framework Figure 1, composed of a database server, in addition, models, views, which are based on a template, which is then visualized through a web browser. The browser sends a request. The view interacts with the model to obtain data (Forcier et al., 2008).

The view calls the template. The template renders the response to the browser request. (Condori Ayala, 2012)

3.6 MVC Django Framework

3.6.1 Model

The Model represents the data structures where all the information with which the application operates manage behavior and domain data typically the model classes will contain functions that help to return, insert and update information from your database, regardless of the database to use or possible changes (González and Romero, 2012).

Encapsulating an application model into components facilitates debugging, improves quality and encourages reuse of code, and for the changes that the system might suffer in the future.

3.6.2 View

- The information that is presented to the user, the view manages the presentation of the information of the application; Everything related to the user interface, the data necessary for the user to continue interacting with the application (González and Romero, 2012).
- A view will normally be a web page, the view represents the presentation logic of the application. The components of the view extract the current system state of the model and provide the user interface for the protocol being used.

3.6.3 Controller

- It is the one that respond to the events invoked from the view; And in turn will call the corresponding methods the model to process and produce

the necessary answers with the data suitable for the interaction of the application (González and Romero, 2012).

- The Controller serves as an intermediary between the Model, the View and any other resources necessary to process the HTTP request and generate a web page (Hendrix et al., 1993).
- The controller is responsible for receiving the events, determining the processor of the event, invoking the processor and finally causing the generation of the appropriate view.
- The controller is responsible for receiving the events, determining the processor of the event, invoking the processor and finally causing the generation of the appropriate view.

3.7 Location of the Application in the TCP/IP Model Layer

OSI model suggests seven logical layers, may be desirable for the modularity of the architectural implementation of a protocol. The TCP / IP architecture normally consists of 5 layers, there have been multiple discussions, however, it gained a number of ideas even greater than the entire ISO protocols (Forouzan, 2002).

Typically, the top three levels of the OSI model (Application, Presentation and Session) are simply considered as the application level in the TCP / IP suite. Because TCP / IP does not have a unified session level on which higher levels are supported, these functions are typically performed (or ignored) by user applications. The most notable difference between the TCP / IP and OSI models is the Application level, in TCP / IP some levels of the OSI model are integrated into their Application level (Bentham and Bentham, 2000).

In the **Figure 3**, you can see how the different layers interact for the execution of our application (Recor), so that we identify which process is executed in each layer, from the application layer until the sending of zeros and ones that are transmitted by a physical port destined to a local network, that connects with the web.

In the application layer we have the data that is sent by our application, which then passes to an application layer, where you have TCP / IP protocol confirmations, which are then sent to the network layer, which then passes to the Link layer in frames, which are finally converted into the latter, in electrical pulses (Handel and Sandford II, 1996).

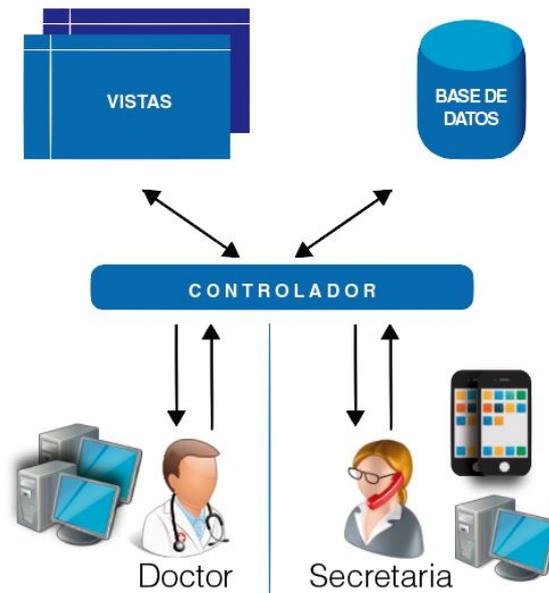


Figure 2: MVC Django Framework.

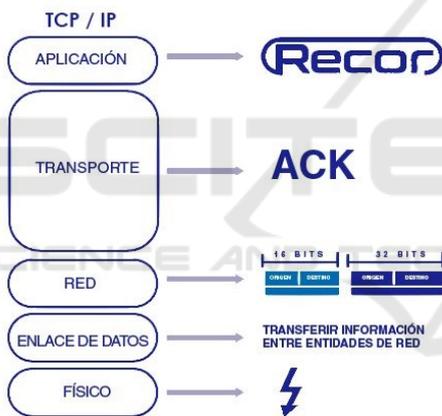


Figure 3: TCP/IP of the application.

4 METHODOLOGY

The use of technologies that allow to separate the different layers present in a Web application and the understanding of its operation within the application to be developed, deserves to be a subject of study. Currently the applications are composed of 3 main layers distributed as follows:

4.1 SCRUM

It is a methodology with an iterative development process that allows constant deliveries to the client, in this case it is an appropriate methodology with a direct collaboration with the user.

Sprints made for the execution of the project and the hours proposed:

S1.- Book Appointments: The user requests the reservation of appointments for his practice (12h).

S2.- User Management: Storage and search of people admitted (15h).

S3.- Booking Block: If there is a previous appointment, block the hours and invalidate their availability (5h).

S4.- Projection of Appointments: To realize a projection interface in screen, for the waiting room (8h).

4.2 Data Modeling

Model.- The model layer that defines the business logic. The database belongs to this layer, which is why we present in the **Figure 4**, the database modeling specific to a reservation system. Where there is a role, so that it is projected in the future that new roles are integrated, such as doctors or patients so that they can access the system, as well as accounting role. It is specified that the receptionist can book one or several appointments and these in turn may contain one or several hours, that is, several periods when an appointment has an extended time (Co-Requisitos and Aprobó,).

View.- The view is what users need to interact with the application (the template managers and pages accessed by the user belong to this layer) In the **Figure 6** all the views are defined that will be occupied for this model (Shaw and Garlan, 1996).

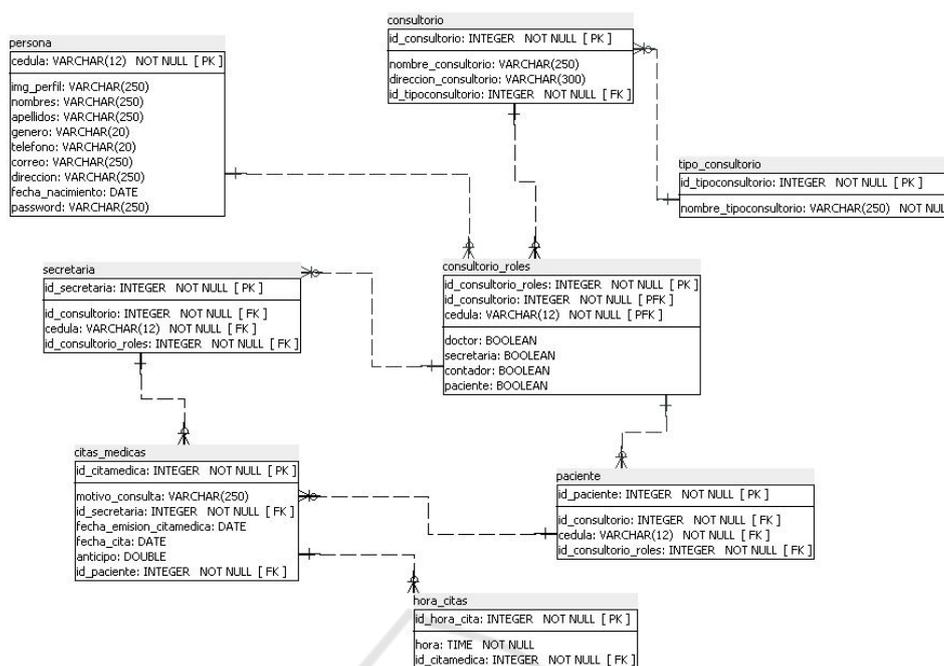


Figure 4: Data Modeling.

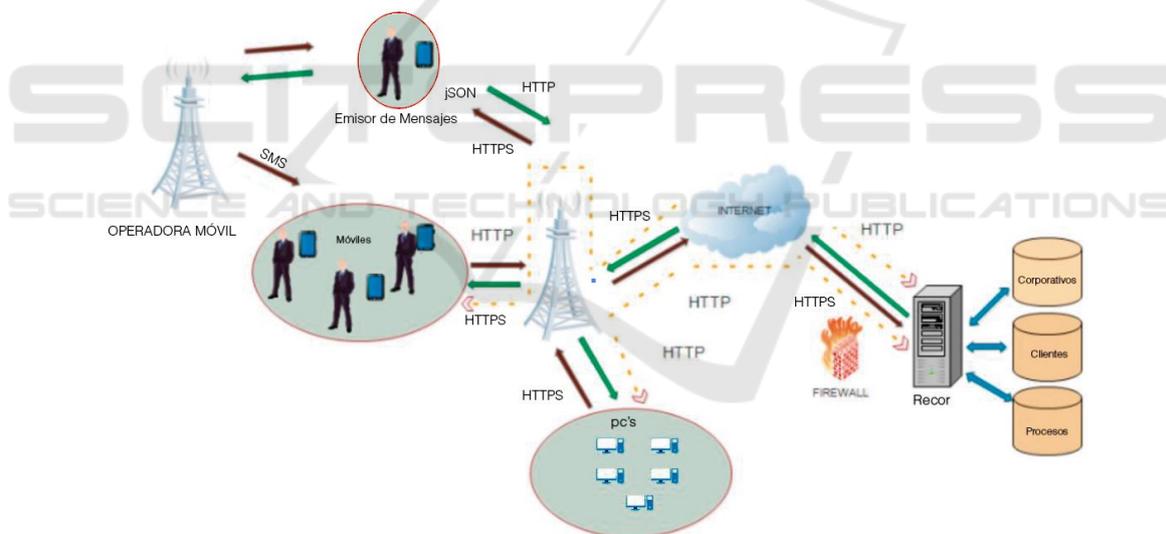


Figure 5: Operative Model.

Controller.- The controller is a block of code that makes calls to the model to obtain the data and passes them to the view showing them to the user. These front controllers actually delegate all work to the actions, and the logical groupings of actions are called modules.

Once explained how our application is structured, we proceed to detail, at the network level how the protocol used by our application in a conventional device.

4.3 Operative Model

The operational model of an application, allows us to see the global picture of its operation, so that it can determine if its operation is coupled with the objectives set at the beginning (Enríquez et al., 2007).

In the **Figure 5**, it is observed in a general level the operation of the application that once the TCP / IP protocols are executed, until a router that then passes to an antenna that connects to the network through an http protocol, which allows us to connect to the

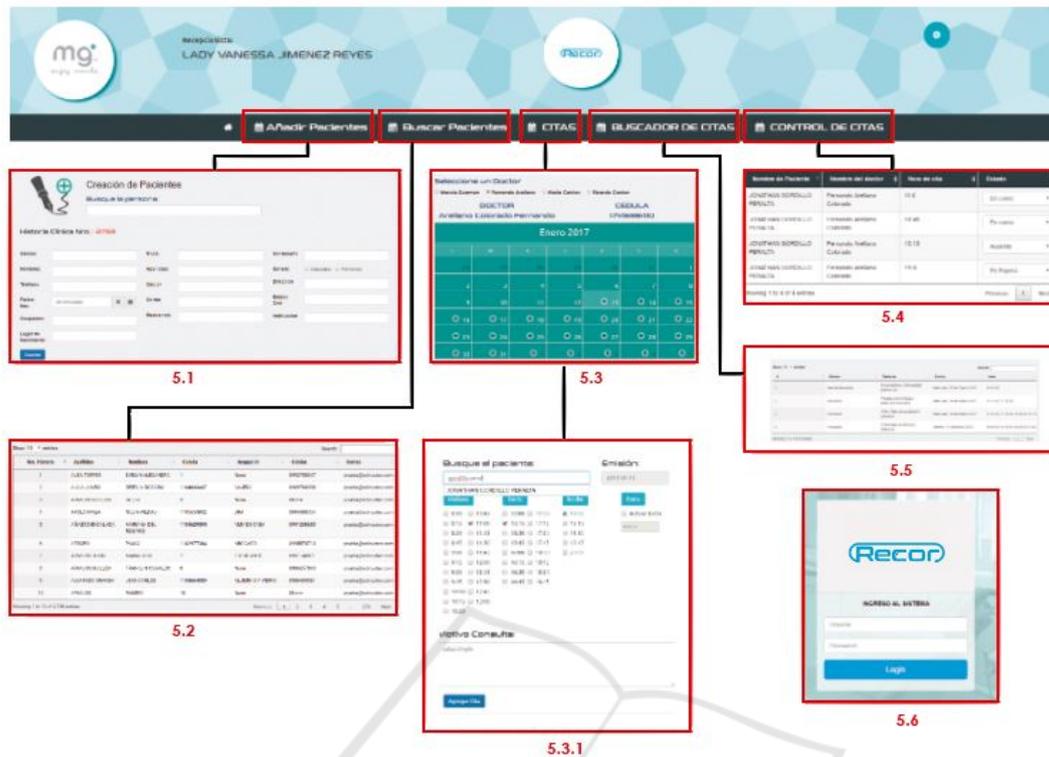


Figure 6: Application.

internet and then access our server, where both the customer data and the data of the professionals who use the application are stored, as well as the data that will be used for run the application processes.

It also explains that for text messaging, a web service is used to retrieve information from the internet with the http protocol through a standard format, such as JSON that allows loading data into an android application, in this case, For this to function as a text messaging sender to proceed with sending the personalized message with the time and date of the next appointment of the patient (Crockford, 2006).

Firewalls are defined, being a web application the main protocol that is occupied is the http protocol, with which it connects to the internet and later to the server with the previously defined modeling (Ioannidis et al., 2000).

5 APPLICATION

The application is developed for reservation of appointments, through the web, which allows the secretary of a specific office, manage patients, through a crud to add / modify / remove patients, so that we can make use Of these data for later reservation in the appointment calendar (López, 2007).

In the **Figure 6**, the general functionalities of the application, which quantitatively represent 5 functionalities, are displayed: Add / Modify Patients, Patient Finder, Appointment Management, Appointment Finder and Control of appointments, so that each of them facilitates and streamlines the work of a receptionist, as well as a friendly and intuitive interface, which allows patient management and appointment reservation, these functionalities are deployed later.

Figure 7: Creating Patients.

In the **Figure 7**, we can see how the application presents an interface to add new patients with an automatically generated history number, where we can store various data that can be used in other processes.

The application has a calendar (**Figure 8**), which allows you to select the doctor to which the medical



Figure 8: Reserve Calendary.

appointment is to be added, so that they are stored independently for each doctor, This data can be sent via text message and e-mail. It is important to note that if there is already an appointment at a certain time it is no longer possible to reserve another appointment at the same time, as the application blocks the appointments picked already.



Figure 9: Daily Reserves.

The stored data is retrieved to store new citations (Figure 9), through a search box that allows access to this data, either by patient’s name or their ID number, At the time of scheduling a new appointment this is sent to the email and via text message.

Quotations can be easily removed through a quick interface that immediately removes registered appointments with a single click (Figure 10), which alters the database by deleting the records and returning the availability Of that hour, so that new reserves can be added. For visualization it can be done through the



Figure 10: Eliminar/Visualizar Citas.

same calendar, or through the application’s search tables, which allows you to search for citations by date, cédula number, names, which thanks to the combination of django with javascript, is allowed Perform dynamic searches through tables.



Figure 11: Delete/Visualize.

The appointments finally arrive at the recipient, through e-mail (Figure 11), and text messaging, thanks to the backend work done by the application at the time of reservations.

6 CONCLUSIONS

- The web application for the online administration of medical appointments has to provide in a fast, transparent and secure way the administration of medical appointments by correctly managing the different user accounts according to the established profiles increasing the security and integrity of the information.
- To send text messaging you have to standardize the data either through json or xml so that you can generate a webservice where you can retrieve this data.
- Determining an operational model is essential, in order to recognize the protocols through which the data passes and are transferred and stored.
- The use of Ubuntu in this case or any open source operating system, is essential for the execution of the project due to the freedom of permissions for

both ports, as well as modification of source files (Lei et al., 2017).

REFERENCES

- Álvarez-García, R., Blasco-Fontecilla, H., Legido-Gil, T., López-Castromán, J., Montoya-Ferrer, A., and Baca-García, E. (2011). Los calendarios electrónicos como herramientas de organización de los servicios de medicina interna. *Revista Clínica Española*, 211(2):98–101.
- Arenburg, R. T., Barillaud, F., Cobb, B. L., and Dutta, S. (2010). Method, system and computer program product for enabling electronic chat with online calendar invitees. US Patent 7,739,340.
- Bentham, J. and Bentham, J. (2000). *TCP/IP lean: web server for embedded systems*.
- Co-Requisitos, C. C. P.-R. and Aprobó, E. R. Base de datos. *Sistema*, 2(4):6.
- Condori Ayala, J. L. (2012). Python-djangoframework de desarrollo web para perfeccionistas basado en el modelo mtv. *Revista de Información, Tecnología y Sociedad*, page 36.
- Crockford, D. (2006). The application/json media type for javascript object notation (json).
- Delegido, D. F. (2016). Aplicaciones de mensajería para dispositivos móviles basadas en comunicación punto a punto local.
- Douglas, K. and Douglas, S. (2003). *PostgreSQL: a comprehensive guide to building, programming, and administering PostgreSQL databases*. SAMS publishing.
- Enciso-Quispe, L., Chavez, E., Sanmartin, C., Quezada-Sarmiento, P. A., and Zelaya-Policarpo, E. (2018a). Peluk: Geolocation of hairdressers under the scrum methodology. In *2018 13th Iberian Conference on Information Systems and Technologies (CISTI)*, pages 1–6.
- Enciso-Quispe, L., Moreno, S., Yaguana, J., Zelaya-Policarpo, E., and Quezada-Sarmiento, P. A. (2018b). A geolocation model of health centers specialized in pediatrics and family medicine using the concept of mobile network. In *2018 13th Iberian Conference on Information Systems and Technologies (CISTI)*, pages 1–6.
- Enciso-Quispe, L., Sarmiento, S., and Zelaya-Policarpo, E. (2018c). Personalized medical alert system based on internet of things with dhis2. In *2018 13th Iberian Conference on Information Systems and Technologies (CISTI)*, pages 1–6.
- Enríquez, T., Jacqueline, L., Cadpata, B., and Patricia, M. (2007). Desarrollo de un sistema para la publicación de horarios de atención y reserva de citas médicas basado en dispositivos de telefonía móvil.
- Erl, T. (2004). *Service-oriented architecture: a field guide to integrating XML and web services*. Prentice Hall PTR.
- Forcier, J., Bissex, P., and Chun, W. J. (2008). *Python web development with Django*. Addison-Wesley Professional.
- Forouzan, B. A. (2002). *TCP/IP protocol suite*. McGraw-Hill, Inc.
- Gallardo, J., López, K., and Gonzales, C. (2007). *Perú: Evolución del Acceso, la Cobertura y la Penetración en los Servicios de Telefonía*. OSIPTEL.
- González, Y. D. and Romero, Y. F. (2012). Patrón modelovista-controlador. *Revista Telemática*, 11(1):47–57.
- Gralla, P. and Lindley, E. (2006). *Cómo funcionan las redes inalámbricas*.
- Hammer, M. and McLeod, D. (1978). The semantic data model: A modelling mechanism for data base applications. In *Proceedings of the 1978 ACM SIGMOD International Conference on Management of Data, SIGMOD '78*, pages 26–36, New York, NY, USA. ACM.
- Handel, T. G. and Sandford II, M. T. (1996). Hiding data in the osi network model. In *International Workshop on Information Hiding*, pages 23–38. Springer.
- Harb, H., Farahat, H., and Ezz, M. (2008). Secure-smspay: secure sms mobile payment model. In *Anti-counterfeiting, Security and Identification, 2008. ASID 2008. 2nd International Conference on*, pages 11–17. IEEE.
- Hassan, Y., Martín Fernández, F. J., and Iazza, G. (2004). Diseño web centrado en el usuario: usabilidad y arquitectura de la información. *Hipertext. net*, (2).
- Hendrix, H. D., Urbanus, P. M., and Asahara, M. (1993). Dmd display system controller. US Patent 5,254,980.
- Ioannidis, S., Keromytis, A. D., Bellovin, S. M., and Smith, J. M. (2000). Implementing a distributed firewall. In *Proceedings of the 7th ACM Conference on Computer and Communications Security, CCS '00*, pages 190–199, New York, NY, USA. ACM.
- Lei, H., Ganjezadeh, F., Jayachandran, P. K., and Ozcan, P. (2017). A statistical analysis of the effects of scrum and kanban on software development projects. *Robotics and Computer-Integrated Manufacturing*, 43:59–67. Special Issue: Extended Papers Selected from FAIM 2014.
- López, C. P. (2007). *Minería de datos: técnicas y herramientas*. Editorial Paraninfo.
- Martel, A. (2014). *Gestión práctica de proyectos con Scrum: Desarrollo de software ágil para el Scrum Master*, volume 1. Antonio Martel.
- McDonald, A., Haslam, C., Yates, P., Gurr, B., Leeder, G., and Sayers, A. (2011). Google calendar: A new memory aid to compensate for prospective memory deficits following acquired brain injury. *Neuropsychological rehabilitation*, 21(6):784–807.
- Ross, J. W., Weill, P., and Robertson, D. C. (2006). *Enterprise architecture as strategy: Creating a foundation for business execution*. Harvard Business Press.
- Shaw, M. and Garlan, D. (1996). *Software architecture: perspectives on an emerging discipline*, volume 1. Prentice Hall Englewood Cliffs.
- Thompson, M. and Antezana, J. (2015). El proceso administrativo.