

Apéndice B

Diagrama de Casos de Uso

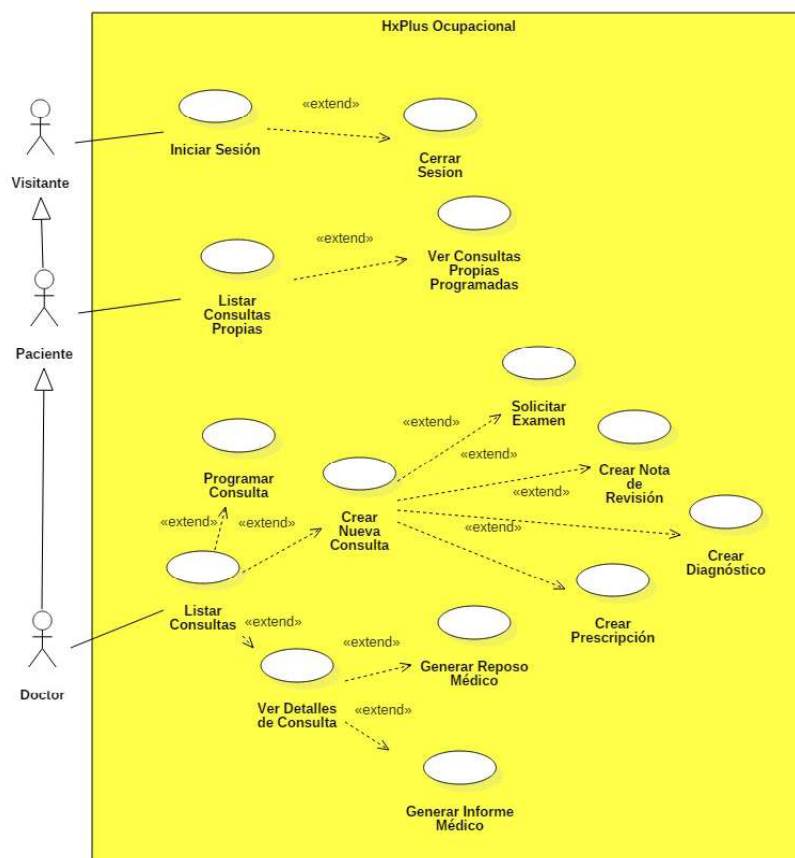


Figura B.1: Diagrama de Casos de uso implementados.

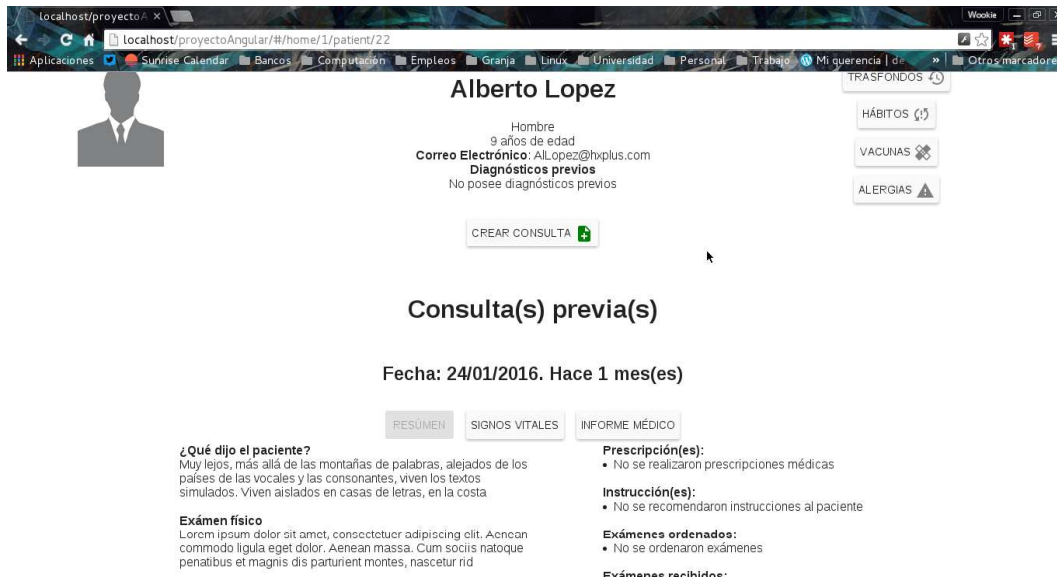


Figura 5.11: Vista de Historial de Consultas Médicas del Paciente

5.2.10 Décimo Sprint: Vista de Revisión de Consulta Médica

Esta vista está insertada en la vista anterior logrando que lo que se llama “Historial Médico” de un paciente sea en sí la lista de todas las consultas médicas junto con los respectivos accesos a trasfondos, alergias, entre otros. La revisión de consulta médica se divide en dos fases:

1. Resumen

Resumen general de la consulta, se muestra la *Soap Note* de la consulta y los exámenes solicitados y recibidos (con su respectivo archivo, según corresponda).

2. Signos Vitales

En esta subvista se muestran los signos vitales recabados durante la consulta. Se muestra un mensaje adecuado en caso de no haberse recabado signos vitales durante dicha consulta. Se hace en un apartado para no saturar la vista principal de “Resumen” con lo que es información muy específica y potencialmente puede ser excesiva para una vista ya cargada.

Para este Sprint se asignaron dos semanas y las pruebas fueron realizadas tomando en cuenta:

- Integración con la vista 5.2.9.
- Actualización de los datos y revisión de los mismos al momento de agregar una nueva consulta (vista 5.2.8).



Figura 5.10: Recepción de Examen Médico

- Respuestas y aceptación de UI.
- Encapsulación adecuada de los datos: Que los datos se preserven entre las distintas subvistas.
- Carga de datos en el sistema: Cantidad de datos y momento de transmisión.
- Formularios dinámicos y la respectiva carga de datos al formulario principal.
- Correctitud de los datos almacenados en el sistema.

Se dejó para el siguiente Sprint la revisión de la correcta carga de datos como la fecha para realizar la prueba conjunta con la revisión de la recuperación de datos de la vista 5.2.9.

5.2.9 Noveno Sprint: Vista de Historial de Consultas Médicas del Paciente

En esta vista se presenta el historial completo de consultas médicas del paciente, ordenadas de manera cronológica y empezando por la última consulta. También se puede acceder a otros aspectos del historial médico, como por ejemplo hábitos, alergias, etc; tal y como se puede ver en la figura 5.11.

La vista permite a su vez la integración con la vista de Revisión de Consulta Médica (ver 5.2.10) y Agregar Consulta (ver 5.2.8).

Las pruebas para esta vista fueron principalmente de aceptación de UI pero se tomó en cuenta la agregación de nuevas consultas a la lista una vez realizada la consulta y almacenada en el sistema. Además se revisó la integración con la vista de agregar consulta médica (ver 5.2.8) y las acciones que ella realiza dentro del sistema.

Para este Sprint se asignaron dos semanas pero debido a problemas de integración con las vistas mencionadas, fue requerido una semana adicional de revisión para el correcto funcionamiento de la vista.

Figura 5.8: Signos Vitales

En este caso se utilizó un formulario dinámico para el registro de solicitud de exámenes médicos. En esta fase, el examen sólo cuenta con un nombre, dado por el doctor tratante, y se registra la solicitud que puede ser recibida en una consulta futura en la sección “Recibir Exámenes” descrita a continuación.

Figura 5.9: Solicitud de Examen Médico

- Recepción de Examen Médico

En esta fase el doctor puede indicar la recepción de un examen médico solicitado en una consulta previa, y sólo de los solicitados en consultas previas, junto con la carga de un archivo asociado al examen (en caso de poseer un archivo digital). La intención es mantener la información médica que llevó a diagnósticos de tal forma que se puedan realizar revisiones a la misma y estudios de evolución de pacientes. Ver figura 5.10.

Para este Sprint, debido a la complejidad del formulario y la encapsulación de los datos requeridos, se asignó una duración de tres semanas en las cuales el equipo de desarrollo realizó la revisión de los tutoriales antes mencionados para lograr la correcta encapsulación de los datos.

Para las pruebas de este Sprint se siguieron los siguientes parámetros:

RESÚMEN GENERAL

SIGNOS VITALES

ORDENAR EXÁMENES

RECIBIR EXÁMENES

¿Qué dice el paciente?

¿Qué se puede observar?

Plan de tratamiento:

Diagnóstico(s):

Agregue un nuevo diagnóstico

Instrucción(es):

Agregue una nueva instrucción

Elija un fármaco registrado
Indicación

Comentarios:

✓

✗

Figura 5.7: Resumen General

vitales tomados en la misma. Se tomó el nombre de “Signos Vitales” por convención aunque refiere a datos fisiológicos en general: altura, peso, presión arterial y cualquier otro que el doctor considere pertinente. Se deja una acción especial para añadir al sistema nuevos parámetros; esto con la finalidad de permitir flexibilidad y mejorar la eficiencia de la carga de datos.

- Flexibilidad: En caso de precisar un nuevo parámetro fisiológico, por no estar previamente contemplado, el doctor puede agregarlo haciendo uso de la opción “Otro”.
- Carga de Datos: Dado que los parámetros requeridos pueden ser diferentes entre las disciplinas médicas, se estaría desperdiciando espacio en la base de datos si se tomara un sólo formulario con todos los parámetros registrados. Tomando en cuenta eso, se decidió que el doctor está en potestad de hacer la revisión del paciente y añadir los “Signos Vitales” que considere pertinentes y que hayan sido medidos durante la consulta; así pues, un mismo doctor puede realizar dos consultas a un mismo paciente y en una registrar la “temperatura corporal” (por ejemplo) y en otra consulta registrar la “tensión arterial”.

- Solicitud de Examen Médico

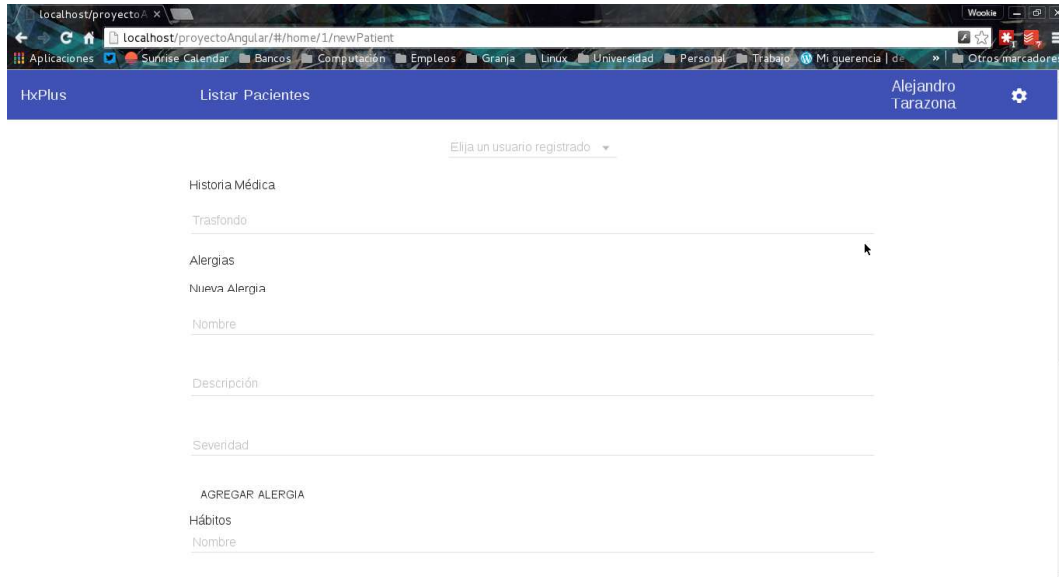


Figura 5.6: Vista de creación de nueva historia médica para un paciente nuevo.

5.2.8 Octavo Sprint: Vista de Agregar Consulta Médica

En esta vista se presentan los formularios respectivos para la creación de una nueva consulta médica separados por subvistas que van llenando cada una de los aspectos requeridos de las consultas médicas tal y como se enumeraron en 5.1.2 y subdivididos en:

- Resumen General:
 - Análisis Subjetivo (*Subjective*)
 - Análisis Objetivo (*Objective*)
 - Plan de tratamiento
 - Diagnóstico(s)

En este caso se hizo uso de un formulario dinámico ya que pueden ser varios diagnósticos realizados en una misma consulta.
 - Instrucción(es)

También en este caso se realizó haciendo uso de los formularios dinámicos. Las instrucciones puede tener una o varias prescripciones médicas asociadas.
 - Comentarios
- Signos Vitales:

Para el caso en que aplique, se tiene esta sección de la consulta donde se registran los signos



Figura 5.4: Vista de Agregar pacientes que ya han sido atendidos por algún médico.

Esta vista se usó como referencia para las pruebas de la “Vista de Agregar Nuevo Paciente” para revisar que la agregación estaba siendo realizada exitosamente.

Para este Sprint se dedicó una semana y las pruebas se basaron en la respuesta de la vista ante los cambios realizados por la vista mencionada y en la aceptación de la UI.

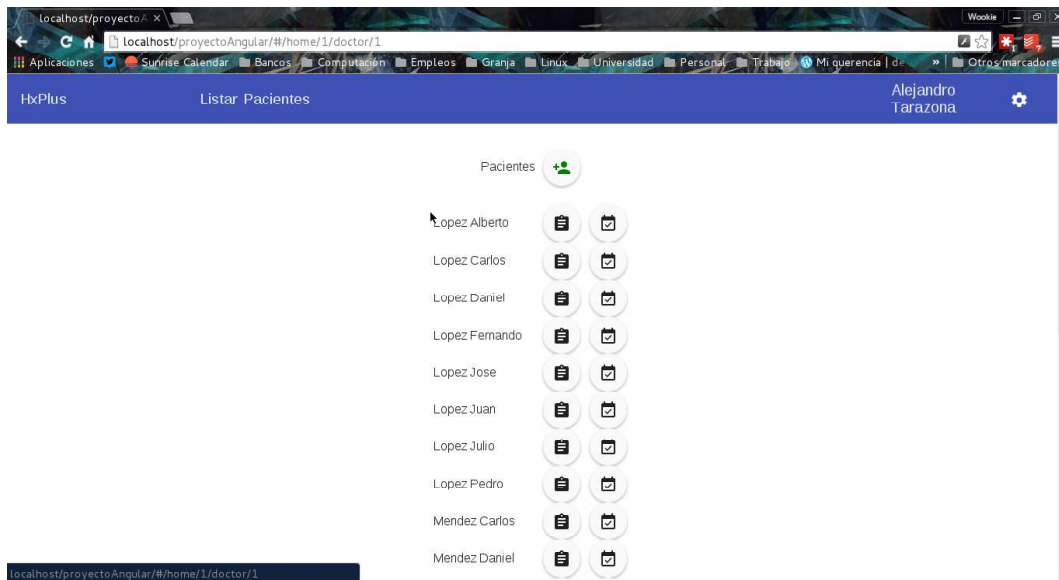


Figura 5.5: Lista de Pacientes del Médico

almacena y regresa a la vista anterior, en caso de éxito, y muestra los datos actualizados del usuario. En caso de algún fallo, el sistema debe dar el mensaje correspondiente al cambio ilegal y mantenerse en la vista con los datos originales del usuario. En caso de modificaciones de la contraseña, el usuario no podrá ver la contraseña previa y el campo estará vacío; este campo se llenará en caso de desear modificarla y se necesita también una verificación de doble escritura de la contraseña para tal fin.

Las pruebas de esta vista fueron:

1. Alterar la información del usuario eliminando campos requeridos (nombre de usuario, contraseña, primer nombre, primer apellido, número de cédula y correo electrónico).
2. Cambiar el nombre de usuario, número de cédula o correo electrónico por valores ya registrados en el sistema.
3. No alterar ningún dato del usuario y enviar el formulario con los datos previos usando el botón de envío de datos.
4. Verificar que los datos fuesen actualizados en la vista anterior en caso de haber verificaciones válidas en el usuario dado.

A partir del caso 2 se detectó que el sistema realizaba una modificación completa del usuario en la base de datos con los datos enviados en el formulario, lo cual podría devenir en problemas de procesamiento a futuro con grandes cargas de datos y es innecesario en el caso 3 por la misma naturaleza de la no modificación. Se acordó, en el siguiente Sprint, revisar esta funcionalidad.

The screenshot shows a web browser window with the URL `localhost/proyectoAngular/#/home/1/userForm/1`. The browser's address bar and tabs are visible at the top. The main content area displays a form for editing a user. The form fields are as follows:

- Nombre de Usuario: atarazona
- Nombre: Alejandro
- Apellido: Tarazona
- Cédula de Identidad: (empty)
- Número de RIF: (empty)
- Número de Teléfono: (empty)
- Correo Electrónico: atarazona@hxplus.com
- Dirección de habitación: Barquisimeto

At the bottom of the form, there is a button labeled "GUARDAR CAMBIOS".

Figura 5.3: Pantalla de Edición de Usuario.

las pruebas de la vista de “Detalles de Usuario”. Se plantea que a futuro sea utilizada por usuarios especiales dentro del sistema y que manejen dichas listas dentro de una empresa.

A este Sprint se le asignó una semana de lapso para su entrega dada la importancia relativamente baja de la vista no se refinó y su UI está muy poco refinada.

Las pruebas de la vista se basaron en el orden de aparición de los usuarios (ordenados alfabéticamente) y la actualización de la lista al agregar un nuevo usuario.

5.2.3 Tercer Sprint: Vista de Detalles de Usuario

Esta vista es la muestra de los datos ingresados al sistema del usuario elegido.

A este Sprint, debido a su simpleza, se le dedicó sólo una semana de tiempo para realizar las pruebas necesarias.

Por ser una vista sencilla, las pruebas se realizaron correlacionando los datos de los usuarios con lo que se muestra en pantalla y la aceptación de UI por parte del *product owner*.

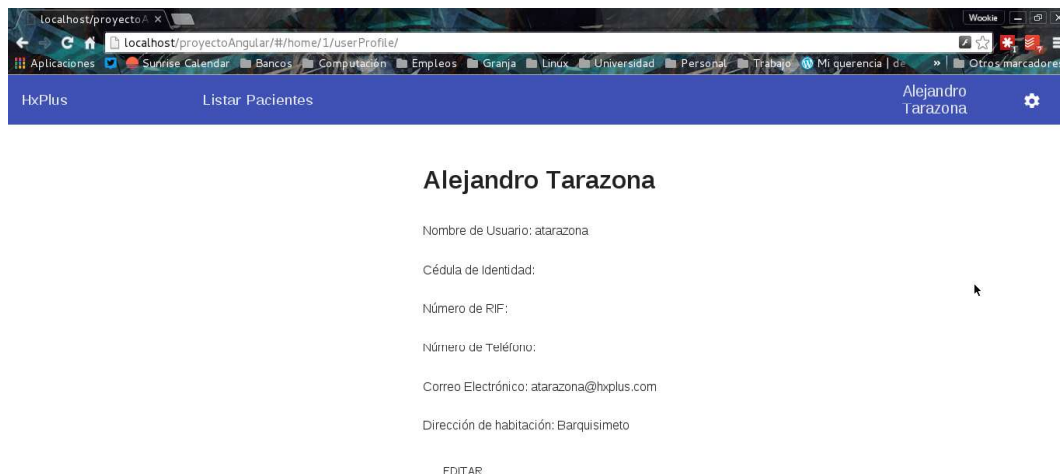


Figura 5.2: Pantalla de Revisión de Usuario.

5.2.4 Cuarto Sprint: Vista de Edición de Usuario

Esta vista muestra un formulario, similar al formulario de registro de usuario, lleno previamente con los datos previamente registrados. Si el formulario cambia y estos cambios son enviados, el sistema los

5.2.1 Primer Sprint: Vista de Autenticación de Usuarios

Se implementó la funcionalidad de autenticación siguiendo los parámetros de autenticación basada en *tokens*, para ello se agregó al “pom.xml”³ las dependencias requeridas para la autenticación. Ver cuadro 5.4 La clave usada por el servidor fue una clave generada en tiempo de ejecución para que la misma fuera cambiante y mejorar la seguridad. Sin embargo, la clave, una vez generada se mantiene igual mientras el servidor esté en funcionamiento. Ver figura 5.1.

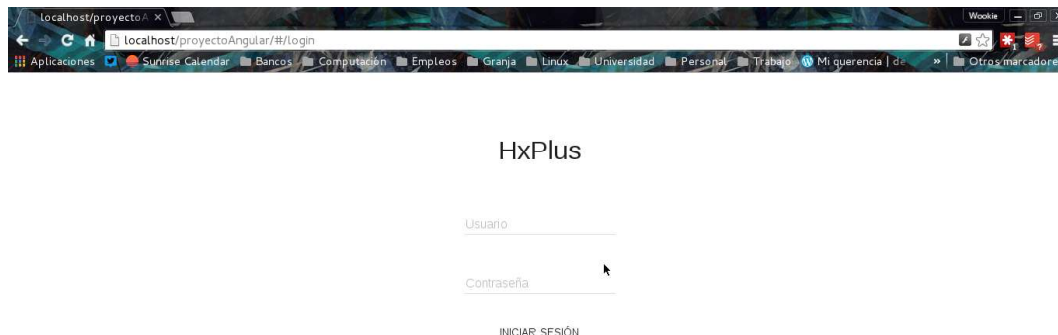


Figura 5.1: Pantalla de Autenticación de usuarios

Grupo	Artefacto	Versión
io.jsonwebtoken	.jjwt	0.5.1

Cuadro 5.4: Artefactos de Maven: Autenticación

Las pruebas sobre esta vista fueron realizadas evaluando la generación del *token* de parte del servidor en *back end* y su posterior uso para el acceso a las demás páginas del sistema. La ficha fue generada con éxito y verificado el cambio de ficha en cada inicio de sesión, se dió la aprobación de la vista para proseguir al siguiente Sprint.

5.2.2 Segundo Sprint: Vista de Lista de Usuarios

Esta vista contiene una lista con los nombres de los usuarios registrados en el sistema. Para el alcance del sistema se implementó una lista general sin discriminaciones sobre dicha lista y fue utilizada en

³Ver punto 4.1.11

4.1.10 SPRING

Framework para el desarrollo de aplicaciones que provee inversión de control; es de código abierto y está diseñado sobre Java. Permite integración con Hibernate, JPA y JSON[26].

SPRING fue diseñado para facilitar el desacoplamiento de los componentes del sistema utilizando IOC. Esto permite que los componentes sean desarrollados una y sólo una vez y que puedan ser reutilizados en diferentes contextos[27].

Su fácil integración con Hibernate y, por consecuencia, con JPA permite que la interacción con la base de datos sea transparente al desarrollador y evita que tenga que reescribirse el código en caso de cambios en el manejador (de bases de datos) utilizados.

4.1.11 Maven

Es una herramienta de gestión y manejo de librerías, parecida a “Apache Ant”. Utiliza el concepto del “Modelo del Objeto de Proyecto” (del inglés *Project Object Model*, o POM) para gestionar la construcción del proyecto dónde se utilice. Esto es, gestiona las librerías, dependencias y versiones (de las librerías) de forma centralizada y limpia.

El POM es un archivo en formato XML dónde se registran las librerías que serán usadas por el proyecto para que el manejador de Maven se encargue de la descarga de las mismas. Se gestiona a través de artefactos que registran la información de una librería de acuerdo con la figura 4.2.



Figura 4.2: Artefacto de Maven y descripción de su contenido.

La estructura del POM puede llegar a ser tan compleja como el proyecto que gestiona, llegando incluso a depender de otros POM. En “HxPlus Ocupacional” se manejó usando un sólo POM de la

3.2 Eventos

Esta sección describe los eventos determinados por la metodología seleccionada y cómo fueron establecidos para la gestión del proyecto. Estos son:

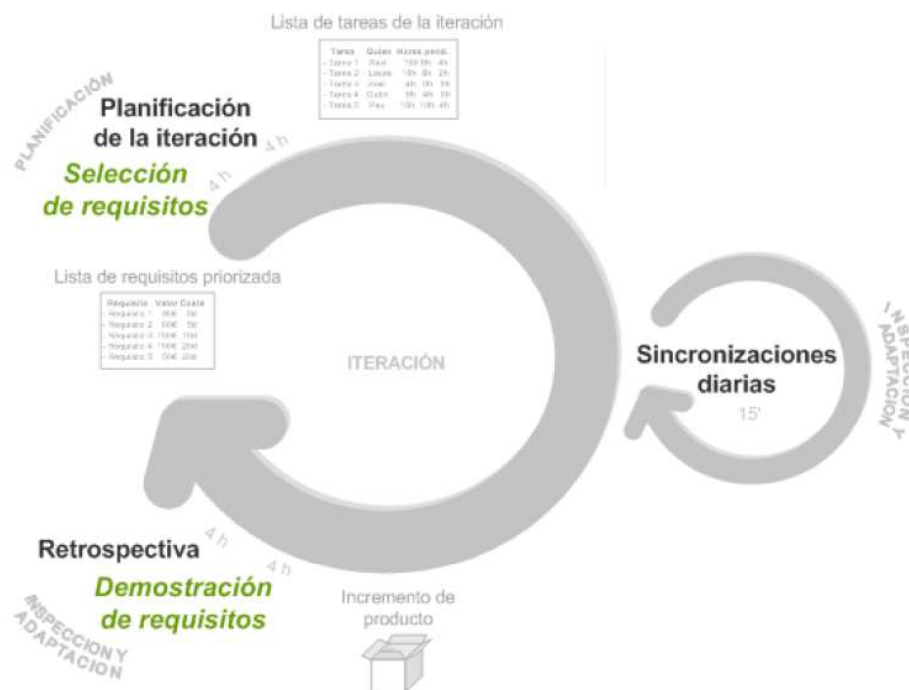


Figura 3.1: Esquema de trabajo de SCRUM

3.2.1 Sprint

Unidad mínima de desarrollo, usualmente determinada por una tarea corta o un período de tiempo pequeño, usualmente 1 o 2 semanas, nunca más de 30 días; durante el cual el equipo de desarrollo trabaja según las metas estipuladas al principio del mismo. Normalmente éstas metas no cambian durante el desarrollo del Sprint sino al final del mismo, cuando se planifica el siguiente Sprint.

Para HxPlus Ocupacional fue determinado en una semana para las primeras tareas y dos para las últimas, debido a las pruebas subyacentes y el trabajo de integración que representan.

Se llevó a cabo 3 fases:

1. Preparación
2. Implementación

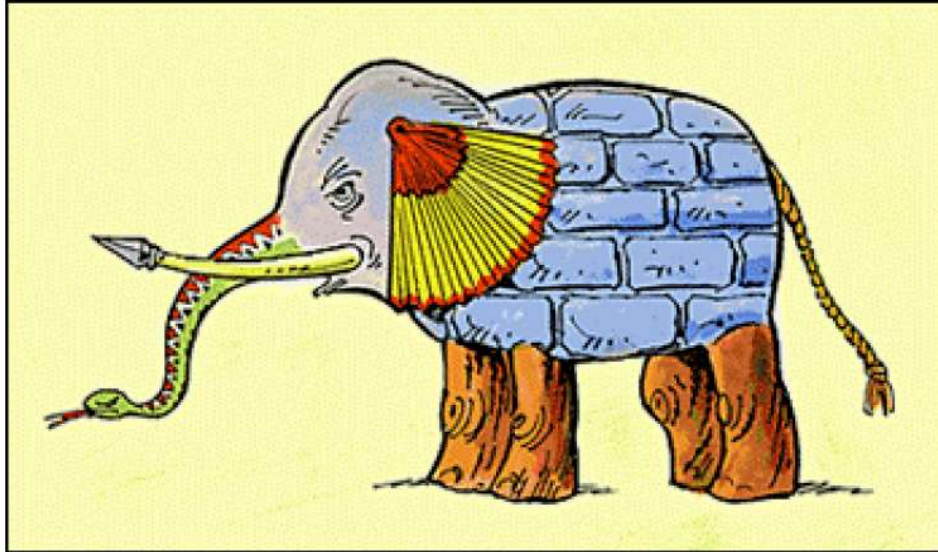


Figura 2.2: El elefante de Saxe

Se usa la analogía para ejemplificar el hecho de que haya varias definiciones diferentes de lo que es SOA en si, porque se le ha definido como patrón de diseño o como una filosofía de desarrollo, siendo esta última la definición adoptada para el trabajo descrito en el presente informe.

También se puede usar dicha analogía para ejemplificar cómo los (potencialmente diversos) dispositivos pueden interactuar con el controlador y este a su vez con el modelo sin que ello represente un cambio fundamental en diseño o estructura de los mismos. Cada dispositivo interactúa con los servicios que necesita y más ningún otro.

2.3 Autenticación Basada en *Tokens*

Autenticación, según se lee en [6], es la “acción y efecto de autenticar”. Esto es comprobar ante una autoridad la veracidad o legitimidad de un documento o un hecho[7].

En sistemas, el término es utilizado para definir el proceso de verificación de las credenciales de usuarios dentro de un sistema. Comunmente usando el par “nombre de usuario” y “contraseña” para ello.

Existen diversos métodos para llevar a cabo la autenticación[8]:

1. Sistemas basados en algo conocido, ya sea *password* o *passphrase*.
2. Sistemas basados en algo poseído que puede ser un tarjeta de identidad o dispositivos USB.

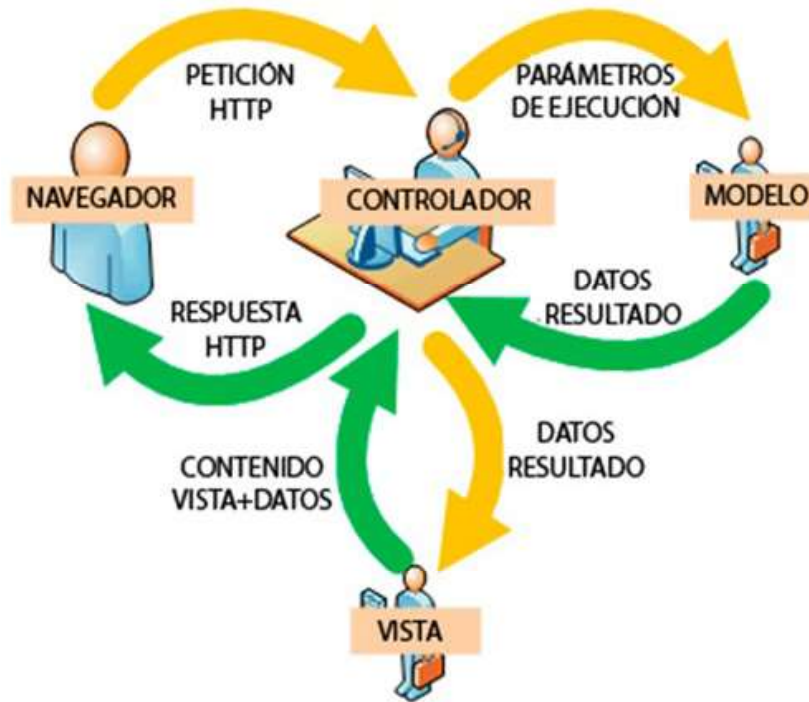


Figura 2.1: Representación gráfica del patrón Modelo Vista Controlador[2]

2. Vista

- Presentación del modelo.
- Accede al modelo pero no puede cambiar su estado.

3. Controlador

- Reacciona a las peticiones del cliente.
- Comunica al modelo de las acciones ejecutadas.
- Direcciona a las vistas requeridas del lado del cliente.

Esto se busca, primordialmente, para hacer del código altamente mantenible en el tiempo, ya que antiguamente se realizaban los sistemas siguiendo lo que se conoce como *“programación de espagueti”* (programación no estructurada) la cual no llevaba una separación entre lo que era la vista y los procesos internos del sistema. Esto traía como consecuencia, en el momento de realizar algún cambio al sistema ya sea de formato o de interacción, que tuviera que modificarse integralmente vista, interacciones y (potencialmente) el modelo de datos.

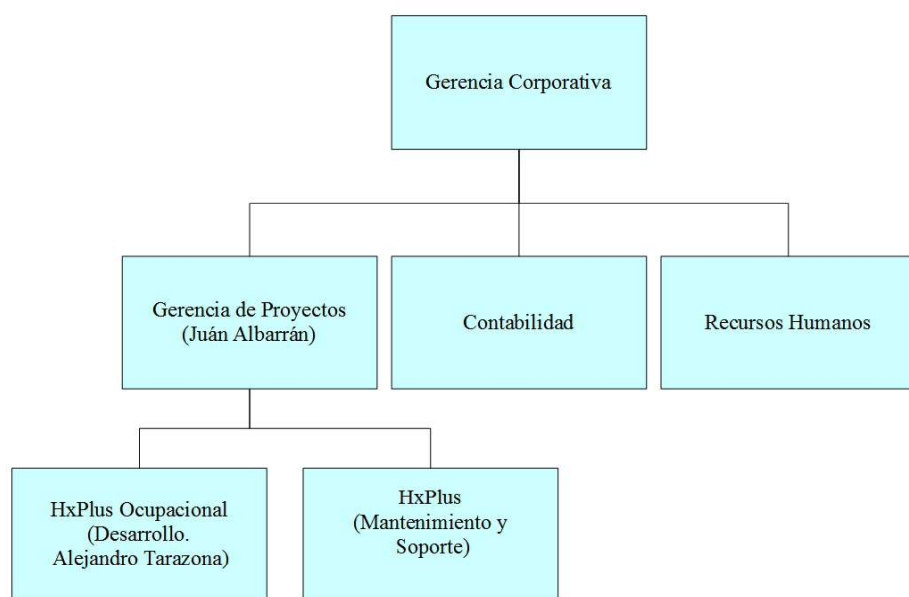


Figura 1.1: Estructura Organizacional de Globinsoft