



# UTMACH

FACULTAD DE INGENIERÍA CIVIL

CARRERA DE INGENIERÍA DE SISTEMAS

ANÁLISIS DE LAS ESTRATEGIAS DE LA METODOLOGÍA SNAIL PARA  
EL DESARROLLO DE SOFTWARE WEB

VINCES SANCHEZ LUIS FERNANDO  
INGENIERO DE SISTEMAS

MACHALA  
2019



# UTMACH

FACULTAD DE INGENIERÍA CIVIL  
CARRERA DE INGENIERÍA DE SISTEMAS

ANÁLISIS DE LAS ESTRATEGIAS DE LA METODOLOGÍA SNAIL  
PARA EL DESARROLLO DE SOFTWARE WEB

VINCES SANCHEZ LUIS FERNANDO  
INGENIERO DE SISTEMAS

MACHALA  
2019



# UTMACH

FACULTAD DE INGENIERÍA CIVIL

CARRERA DE INGENIERÍA DE SISTEMAS

EXAMEN COMPLEXIVO

ANÁLISIS DE LAS ESTRATEGIAS DE LA METODOLOGÍA SNAIL PARA EL  
DESARROLLO DE SOFTWARE WEB

VINCES SANCHEZ LUIS FERNANDO  
INGENIERO DE SISTEMAS

MOLINA RIOS JIMMY ROLANDO

MACHALA, 21 DE AGOSTO DE 2019

MACHALA  
21 de agosto de 2019

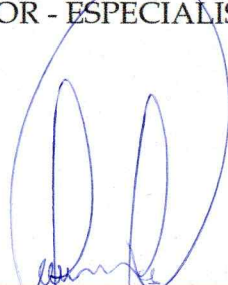
**Nota de aceptación:**

Quienes suscriben, en nuestra condición de evaluadores del trabajo de titulación denominado ANÁLISIS DE LAS ESTRATEGIAS DE LA METODOLOGÍA SNAIL PARA EL DESARROLLO DE SOFTWARE WEB, hacemos constar que luego de haber revisado el manuscrito del precitado trabajo, consideramos que reúne las condiciones académicas para continuar con la fase de evaluación correspondiente.



---

MOLINA RIOS JIMMY ROLANDO  
0703691980  
TUTOR - ESPECIALISTA 1



---

HONORES TAPIA JOOFRE ANTONIO  
0704811751  
ESPECIALISTA 2



---

CÁRDENAS VILAVICENCIO OSCAR EFREN  
0703935312  
ESPECIALISTA 3

Fecha de impresión: miércoles 21 de agosto de 2019 - 15:10

## Urkund Analysis Result

**Analysed Document:** Luis Fernando Vincés Sanchez.docx (D54750516)  
**Submitted:** 8/8/2019 11:52:00 PM  
**Submitted By:** jmolina@utmachala.edu.ec  
**Significance:** 4 %

Sources included in the report:

[https://resources.sei.cmu.edu/asset\\_files/WhitePaper/2010\\_019\\_001\\_28782.pdf](https://resources.sei.cmu.edu/asset_files/WhitePaper/2010_019_001_28782.pdf)

Instances where selected sources appear:

9



## CLÁUSULA DE CESIÓN DE DERECHO DE PUBLICACIÓN EN EL REPOSITORIO DIGITAL INSTITUCIONAL

El que suscribe, VINCES SANCHEZ LUIS FERNANDO, en calidad de autor del siguiente trabajo escrito titulado ANÁLISIS DE LAS ESTRATEGIAS DE LA METODOLOGÍA SNAIL PARA EL DESARROLLO DE SOFTWARE WEB, otorga a la Universidad Técnica de Machala, de forma gratuita y no exclusiva, los derechos de reproducción, distribución y comunicación pública de la obra, que constituye un trabajo de autoría propia, sobre la cual tiene potestad para otorgar los derechos contenidos en esta licencia.

El autor declara que el contenido que se publicará es de carácter académico y se enmarca en las disposiciones definidas por la Universidad Técnica de Machala.

Se autoriza a transformar la obra, únicamente cuando sea necesario, y a realizar las adaptaciones pertinentes para permitir su preservación, distribución y publicación en el Repositorio Digital Institucional de la Universidad Técnica de Machala.

El autor como garante de la autoría de la obra y en relación a la misma, declara que la universidad se encuentra libre de todo tipo de responsabilidad sobre el contenido de la obra y que asume la responsabilidad frente a cualquier reclamo o demanda por parte de terceros de manera exclusiva.

Aceptando esta licencia, se cede a la Universidad Técnica de Machala el derecho exclusivo de archivar, reproducir, convertir, comunicar y/o distribuir la obra mundialmente en formato electrónico y digital a través de su Repositorio Digital Institucional, siempre y cuando no se lo haga para obtener beneficio económico.

Machala, 21 de agosto de 2019



VINCES SANCHEZ LUIS FERNANDO  
0707086484

## **DEDICATORIA**

A Dios, por concederme salud, fortaleza e inteligencia para cumplir mis objetivos personales, laborales y académicos.

A mis familiares, principalmente, a mis padres, por su ayuda incondicional, consejos y ejemplo de superación.

A mi hija, por llenarme de alegría y felicidad, convirtiéndose en mi inspiración.

A los docentes, por brindarme sus conocimientos en cada una de las asignaturas de la carrera académica.

Sr. Vines Sánchez Luis Fernando

## **AGRADECIMIENTO**

A Dios, por otorgarme salud, protección y sabiduría para lograr mis objetivos.

A mi familia, especialmente a mis padres, que han estado conmigo en todo momento. Además de su apoyo económico, me han inculcado buenos principios orientándome siempre por el camino del bien.

A mis compañeros y amigos, por las experiencias compartidas en el lapso de la carrera académica.

A mi tutor, Ing. Jimmy Molina, que mediante su experiencia y amplios conocimientos dirigió el desarrollo de la parte práctica de mi examen complejo.

A los docentes, por las clases impartidas, siendo estas la base fundamental para mi desarrollo académico y profesional.

A la Universidad Técnica de Machala, Facultad de Ingeniería Civil y la carrera de Ingeniería de Sistemas, por darme la oportunidad de prepararme profesionalmente en el área de sistemas.

Sr. Vences Sánchez Luis Fernando



## RESUMEN

### ANÁLISIS DE LAS ESTRATEGIAS DE LA METODOLOGÍA SNAIL PARA EL DESARROLLO DE SOFTWARE WEB

Vinces Sánchez Luis Fernando, 0707086484

Actualmente es evidente la expansión del internet y el uso de software web, este último ha traído múltiples beneficios que satisface las necesidades de la sociedad en diferentes índoles, al mismo tiempo que incrementa la dependencia a esta tendencia, surge la obligación de utilizar métodos sistemáticos y especializados para su desarrollo. La metodología híbrida SNAIL (Software Nativo de Arquitectura Iterativa Lógica), en efecto, es el análisis y combinación de diversas ventajas implementadas por los procedimientos existentes, este nuevo enfoque ha causado expectativas positivas como alternativa ante las metodologías tradicionales y ágiles, sin embargo, al tratarse de un método emergente, hasta el momento no se han aplicado estándares, normas o modelos de evaluación para valorar su proceso. En base a lo mencionado, el presente trabajo tiene como fin analizar las estrategias SNAIL, mediante los objetivos específicos que establece CMMI-DEV (Modelo de Madurez de Capacidades de Integración para el desarrollo, Capability Maturity Model Integration for Development) nivel 2 y los criterios de la ingeniería web, permitiendo a las personas u organizaciones conocer y aplicar esta metodología para la creación de software web e incluso en condiciones de obtener una certificación CMMI.

**Palabras Claves:** CMMI-DEV, estrategias, ingeniería web, metodología, SNAIL, software web.

## ABSTRACT

### ANALYSIS OF THE STRATEGIES OF THE SNAIL METHODOLOGY FOR THE DEVELOPMENT OF WEB SOFTWARE

Vinces Sánchez Luis Fernando, 070708484

Currently the expansion of the internet and the use of web software is evident, the latter has brought multiple benefits that meet the needs of society in different areas, while increasing dependence on this trend, the obligation to use systematic methods and specialized for its development. The hybrid methodology SNAIL (Native Software of Logical Iterative Architecture), in effect, is the analysis and combination of various advantages implemented by existing procedures, this new approach has caused positive expectations as an alternative to traditional and agile methodologies, however, it is an emerging method, so far standards, norms or evaluation models have not been applied to evaluate its process. Based on the aforementioned, this paper aims to analyze the SNAIL strategies, through the specific objectives established by CMMI-DEV (Capability Maturity Model Integration for Development, Capability Maturity Model Integration for Development) level 2 and the criteria of web engineering, allowing people or organizations to know and apply this methodology for the creation of web software and even in conditions to obtain a CMMI certification.

**Key Words:** CMMI-DEV, strategies, web engineering, methodology, SNAIL, web software.

## ÍNDICE DE CONTENIDO

|  |    |
|--|----|
| DEDICATORIA .....                                  | 1  |
| AGRADECIMIENTO .....                               | 2  |
| RESUMEN .....                                      | 3  |
| ABSTRACT .....                                     | 4  |
| 1. INTRODUCCIÓN .....                              | 7  |
| 1.1 MARCO CONTEXTUAL .....                         | 8  |
| 1.2 PROBLEMA .....                                 | 8  |
| 1.3 OBJETIVO GENERAL .....                         | 8  |
| 2. DESARROLLO .....                                | 9  |
| 2.1 MARCO TEÓRICO .....                            | 9  |
| 2.1.1 SOFTWARE WEB .....                           | 9  |
| 2.1.2 METODOLOGÍAS DE DESARROLLO DE SOFTWARE ..... | 9  |
| 2.1.3 METODOLOGÍAS TRADICIONALES .....             | 9  |
| 2.1.4 METODOLOGÍAS ÁGILES .....                    | 9  |
| 2.1.5 METODOLOGÍAS HÍBRIDAS .....                  | 9  |
| 2.1.6 METODOLOGÍA SNAIL .....                      | 10 |
| 2.1.7 MODELO CMMI Y CMMI-DEV .....                 | 10 |
| 2.1.8 MODELO CMMI-DEV NIVEL DE MADUREZ 2 .....     | 10 |
| 2.1.9 INGENIERÍA WEB .....                         | 11 |
| 2.2 MARCO METODOLÓGICO .....                       | 11 |
| 2.3 RESULTADOS .....                               | 13 |
| 3. CONCLUSIONES .....                              | 17 |
| REFERENCIAS BIBLIOGRÁFICAS .....                   | 18 |
| ANEXOS .....                                       | 21 |

## CONTENIDO DE TABLAS

|  |    |
|--|----|
| <b>Tabla 1</b> Criterios de la ingeniería web .....                                    | 11 |
| <b>Tabla 2</b> Propósito y objetivos específicos de las áreas CMMI-DEV nivel 2 .....   | 12 |
| <b>Tabla 3</b> Sistema de evaluación en base a CMMI-DEV nivel 2 .....                  | 13 |
| <b>Tabla 4</b> Código de las estrategias SNAIL.....                                    | 13 |
| <b>Tabla 5</b> Estrategias SNAIL en base a los criterios de la ingeniería web. ....    | 13 |
| <b>Tabla 6</b> Análisis de las estrategias SNAIL en base a CMMI-DEV nivel 2 .....      | 14 |
| <b>Tabla 7</b> Estrategias SNAIL.....  | 21 |
| <b>Tabla 8</b> Evaluación de las estrategias SNAIL en el área de proceso – CM.....     | 25 |
| <b>Tabla 9</b> Evaluación de las estrategias SNAIL en el área de proceso – MA .....    | 25 |
| <b>Tabla 10</b> Evaluación de las estrategias SNAIL en el área de proceso – PMC .....  | 26 |
| <b>Tabla 11</b> Evaluación de las estrategias SNAIL en el área de proceso – PP.....    | 26 |
| <b>Tabla 12</b> Evaluación de las estrategias SNAIL en el área de proceso – PPQA ..... | 27 |
| <b>Tabla 13</b> Evaluación de las estrategias SNAIL en el área de proceso – REQM.....  | 27 |
| <b>Tabla 14</b> Evaluación de las estrategias SNAIL en el área de proceso – SAM.....   | 27 |
| <b>Tabla 15</b> RF01 – Gestión de piscinas .....                                       | 28 |
| <b>Tabla 16</b> RF02 – Gestión de especies .....                                       | 28 |
| <b>Tabla 17</b> RF03 – Gestión de aireadores .....                                     | 28 |
| <b>Tabla 18</b> RF04 – Gestión de corridas.....  | 29 |
| <b>Tabla 19</b> RF05 – Gestión de siembras.....  | 29 |
| <b>Tabla 20</b> RF06 – Monitoreo .....   | 29 |

## CONTENIDO DE ILUSTRACIONES

|   |    |
|---|----|
| <b>Ilustración 1</b> Soporte de las estrategias SNAIL en CMMI-DEV nivel 2 .....     | 14 |
| <b>Ilustración 2</b> Esquema del Área de Proceso CMMI-DEV .....                     | 21 |
| <b>Ilustración 3</b> Diseño de la base de datos del software web SHIRMPSystem ..... | 30 |
| <b>Ilustración 4</b> Módulo de gestión de piscinas – Listar .....                   | 30 |
| <b>Ilustración 5</b> Módulo de gestión de especies – Editar .....                   | 31 |
| <b>Ilustración 6</b> Módulo de gestión de corridas – Editar.....                    | 31 |
| <b>Ilustración 7</b> Módulo de gestión de siembras – Registrar.....                 | 31 |
| <b>Ilustración 8</b> Módulo de monitoreo – Visualización de piscinas .....          | 32 |
| <b>Ilustración 9</b> Módulo de monitoreo – Tiempo real.....                         | 32 |

## 1. INTRODUCCIÓN

La mayor parte de la implementación denominada TIC (Tecnologías de la Información y Comunicación) se debe a internet, red que dio apertura a la difusión de la información siendo la principal causa para la creación de la World Wide Web (WWW) y los servicios asociados a la misma. El software web dejó de ser simples páginas informativas transformándose en un producto sofisticado a gran escala, que automatiza las actividades de las organizaciones con comodidad, rapidez y disponibilidad inmediata, hechos fundamentales para cumplir objetivos a mayor alcance [1]. No obstante, esta evolución ha generado requerimientos más complejos lo que ha llevado a considerar métodos para la construcción de software web eficiente.

Los procesos implementados influyen en la planificación, elaboración, control y mantenimiento del producto final [2], por tanto el uso de procedimientos sistemáticos no garantiza el éxito de los proyectos del entorno web ya que un proceso inadecuado genera problemas incontrolables en su ejecución, por ello es sustancial que las empresas evalúen su sistema de trabajo en relación a estándares, normas o modelos de calidad.

SNAIL, es una metodología híbrida compuesta por las ventajas que proveen los grupos tradicionales y ágiles [3], reúne buenas prácticas de las metodologías: RUP (Proceso unificado de Rational, en inglés Rational Unified Process), XP (Programación Extrema, en inglés Extreme Programming), SCRUM y OOHDM (Método de Diseño Orientado a Objetos de Hipermedia, en inglés Object Oriented Hypermedia Design Method) [4], ejecuta una serie de fases y actividades en un ciclo iterativo (modelo en espiral), siendo relevante en la creación de software web.

El presente trabajo, tiene como finalidad analizar las estrategias propuestas por SNAIL, a través del Modelo de Madurez de Capacidades de Integración CMMI-DEV nivel 2 y los criterios de la ingeniería web, lo cual permite la aprobación de una metodología apropiada en el desarrollo de sistemas enfocados a la web. La estructura del trabajo establece 3 capítulos, el capítulo 1: detalla el marco contextual, planteamiento del problema y el objetivo en base al problema establecido; el capítulo 2: contempla los contenidos teóricos pertinentes para el desarrollo del proyecto, así también, presenta el marco metodológico para la solución del problema y los resultados obtenidos; el capítulo 3: concluye en relación al análisis realizado.

## **1.1 MARCO CONTEXTUAL**

Los sistemas de información, desde sus inicios, se fundamentaron como una herramienta importante para las organizaciones [5] y con el progresivo crecimiento del internet, el mercado de estos productos se ha inclinado por la adquisición de software web, debido a las múltiples ventajas que ofrece, cubriendo necesidades esenciales en el sector primario, industrial y de servicios [6].

Aunque, para los desarrolladores esta estricta demanda se vuelve un proceso de construcción complejo, puesto que deben adaptarse a los requisitos cambiantes y demás características particulares que requiere la producción de software web [7]. La aplicación de metodologías tradicionales y ágiles en este tipo de desarrollo no son efectivas, ya que no integran estrategias especializadas para solventar estos aspectos, lo que conlleva a la inversión exagerada de recursos, sobretodo poniendo en riesgo la calidad del producto final [3].

Por tal motivo, es necesario enfocarse en nuevas metodologías y analizar sus prácticas a través de estándares, normas o modelos de calidad, con la finalidad de validar un método que pueda dar soluciones a los inconvenientes presentados en la creación de software web.

## **1.2 PROBLEMA**

La utilización de software web se incrementa constantemente y sus usuarios son cada vez más exigentes, por tanto, los desarrolladores deben examinar las particularidades que solicita la elaboración de esta clase de software para no perjudicar su beneficio ni del cliente. SNAIL es una metodología cuyo proceso es diseñado peculiarmente para el desarrollo de software web, de tal forma que cubre las dificultades de los métodos convencionales que tienen alta probabilidad de fallo.

Sin embargo, es importante identificar y evaluar las estrategias claves que propone, para mitigar el riesgo y alcanzar un producto con éxito. ¿Cuáles serían las estrategias que presenta la metodología SNAIL para el desarrollo de software web?

## **1.3 OBJETIVO GENERAL**

Analizar las estrategias de la metodología SNAIL, para el desarrollo de software web, mediante el modelo CMMI-DEV nivel 2 y la ingeniería web.

## **2. DESARROLLO**

### **2.1 MARCO TEÓRICO**

#### **2.1.1 SOFTWARE WEB**

Es un sistema de información que facilita determinadas funciones por medio de internet, se encuentra alojado en un servidor web donde sus usuarios acceden mediante el uso de un navegador (Google Chrome, Mozilla Firefox, etc) [8]. La naturaleza del software web es la utilización de internet, accesibilidad por parte de múltiples usuarios, rendimiento, disponibilidad, variedad de datos, contenido sostenible, actualización continua, inmediatez, seguridad y estética [9] [10].

#### **2.1.2 METODOLOGÍAS DE DESARROLLO DE SOFTWARE**

Son métodos que establecen procedimientos, técnicas y herramientas para planificar, ejecutar, direccionar y monitorear la producción de software que satisfaga las necesidades del cliente [11] [12]. Estas metodologías han evolucionado, enmarcándose en 3 grupos: tradicionales, ágiles e híbridos [13].

#### **2.1.3 METODOLOGÍAS TRADICIONALES**

Las metodologías tradicionales o pesadas son métodos de trabajo para el proceso de desarrollo del software. Su ciclo es secuencial; controlan el proceso con una rigurosa definición de roles, actividades, herramientas, artefactos y documentación. No son flexibles al cambio, por ello se consideran inadecuadas para escenarios con requisitos variantes [14] [15].

#### **2.1.4 METODOLOGÍAS ÁGILES**

Las metodologías ágiles o ligeras son métodos de desarrollo del software, con principios y técnicas que permiten adaptar la forma de trabajo a las circunstancias que requiera el proyecto [16]. Infunden la colaboración de un equipo multidisciplinario incluyendo al cliente. Su ciclo es iterativo; realizan revisiones y entrega de prototipos en cortos periodos de tiempo aprobando la elaboración de un mejor producto [17]. Aunque, tienen desventajas como dependencia del líder de equipo, escasa documentación, insuficiente control de calidad, etc [18].

#### **2.1.5 METODOLOGÍAS HÍBRIDAS**

Las metodologías híbridas o mixtas son la composición de las mejores prácticas de los métodos tradicionales y ágiles, integran ventajas de ambas creando un proceso simple, entendible, documentado, comprometido con el alcance,



participación del cliente y flexible a los cambios del entorno [19]. Su utilización ha ido incrementando poco a poco, ya que se adaptan a las necesidades de los proyectos, generando mayor productividad. Algunos ejemplos son: EssUP (Essencial Unified Process), SXP (Scrum y XP), SNAIL, etc [20].

#### **2.1.6 METODOLOGÍA SNAIL**

Es una metodología híbrida para el desarrollo de proyectos de software web. Se caracteriza por la planificación incremental, simplicidad de construcción, entregas evolutivas, pruebas constantes, adaptabilidad, participación del cliente y comunicación. Sus objetivos se enfocan en la satisfacción del cliente, trabajo potencial, reducción de riesgo en tiempo, costo, calidad y alcance [21]. Su flujo de proceso está basado en el modelo espiral, de ahí su nombre. Tiene las siguientes fases: requisitos, planificación, diseño, programación, pruebas, clausura e inbound marketing (fase opcional) [4].

#### **2.1.7 MODELO CMMI Y CMMI-DEV**

El Modelo de Madurez de Capacidades de Integración CMMI, integra un conjunto de buenas prácticas para la evaluación y mejora de procesos de desarrollo, mantenimiento, adquisición y operación de sistemas. Es administrado por SEI (Instituto de Ingeniería de Software) de la CMU (Universidad Carnegie Mellon) en Estados Unidos. Posee 5 niveles de madurez: inicial, gestionado, definido, gestionado cuantitativamente y optimizado, los mismos que indican el nivel de proceso que ha logrado una organización relativos a un conjunto de áreas de proceso [22]. Consta de 3 modelos, uno de ellos es CMMI-DEV o CMMI para el desarrollo de proyectos de software, tiene como objetivo mejorar los procesos de ingeniería en organizaciones orientadas a la construcción de software [23].

#### **2.1.8 MODELO CMMI-DEV NIVEL DE MADUREZ 2**

El nivel de madurez 2 o gestionado, indica que los proyectos de la organización han asegurado que los procesos se planifiquen y ejecuten; empleando personal calificado, recursos correctos para producir productos controlados; involucrando a las partes interesadas relevantes; son monitoreados, controlados, revisados y evaluados. La disciplina de este nivel permite garantizar que las prácticas existentes se mantienen en períodos difíciles. Cuando estas prácticas se cumplen, los proyectos se realizan y gestionan según los planes documentados. El estado de los productos y los compromisos determinados entre las partes interesadas se revisan según sea necesario [24].

### 2.1.9 INGENIERÍA WEB

Es una área de la ingeniería de software que hace uso de principios, procesos, técnicas y metodologías sistemáticas para desarrollar con éxito y alta calidad software orientado a la web [14].

## 2.2 MARCO METODOLÓGICO

Para la correspondiente validación de SNAIL como una metodología apropiada para el desarrollo de software web, es necesario identificar y evaluar sus estrategias en base a criterios, normas o modelos de calidad dentro del contexto. La ingeniería web establece criterios que debe cumplir un proceso adecuado para la construcción de este tipo de sistemas de información, estos son: actividades estructurales (AE), flujo de proceso (FP) y elementos de diseño (ED) [19] [25] [26].

**Tabla 1** Criterios de la ingeniería web

| Criterios | Descripción   |
|-----------|---|
| AE        | Los proyectos de software web deben contar con un marco de trabajo genérico conformado por las actividades estructurales de comunicación, planeación, modelado, construcción y despliegue.  |
| FP        | Las actividades estructurales deben ejecutarse en un ciclo iterativo, teniendo en cuenta factores significativos como: requerimientos cambiantes, trabajo en equipo, comunicación, participación y satisfacción del cliente o usuario final.  |
| ED        | La calidad del software web definida en términos de usabilidad, funcionalidad, confiabilidad, eficiencia, facilidad de mantenimiento, seguridad y escalabilidad depende de la etapa de diseño, la misma que debe caracterizarse en sencillez, consistencia, identidad, robustez, navegabilidad y atractivo visual. Para cumplirlo la ingeniería web se centra en 6 diferentes elementos de diseño como es: interfaz, estética o diseño gráfico, contenido, arquitectura, navegación y componentes. Otra alternativa es aplicar métodos para diseño de hipermedia orientado a objetos. |

**Fuente:** Elaboración propia

Por otra parte, la presencia de un proceso no garantiza que el software se entregue a tiempo, satisfaga las expectativas del cliente o tenga las características técnicas para un producto de calidad a largo plazo. Los patrones de proceso deben acoplarse con una práctica sólida, cumpliendo criterios básicos y esenciales para el éxito [25]. Existen numerosos enfoques para la evaluación y mejora del proceso de desarrollo de software, en este caso se evalúa el proceso SNAIL según el soporte que tienen sus estrategias en relación al nivel de madurez 2 del modelo. Dicho nivel está compuesto por 7 áreas de proceso: Gestión de Configuraciones (Configuration Management, CM), Medición y Análisis (Measurement and Analysis, MA), Monitoreo y Control del Proyecto (Project Monitoring and Control, PMC), Planeamiento del Proyecto (Project Planning, PP), Aseguramiento de la Calidad del Proceso y del Producto (Process and Product Quality Assurance, PPQA), Gestión de los Requerimientos (Requirements Management, REQM), Gestión de los Acuerdos con Proveedores (Supplier Agreement Management, SAM). Cada área de proceso,

conduce a satisfacer sus objetivos específicos, estos en prácticas específicas y éstas en subprácticas, (Ver Anexo A).

A continuación, se describe el propósito y los objetivos específicos de cada área de proceso [27]:

**Tabla 2** Propósito y objetivos específicos de las áreas CMMI-DEV nivel 2

| Área | Propósito   | Objetivos Específicos (OE)   |
|------|---|--|
| CM   | Establecer y mantener la integridad de los productos de trabajo utilizando la identificación, control, registro y auditorías de configuración.                        | <b>OE1. Establecer líneas base:</b> se establecen líneas base de productos de trabajo identificados.<br><b>OE2. Seguimiento y control de cambios:</b> los cambios en los productos de trabajo son rastreados y controlados.<br><b>OE3. Establecer integridad:</b> se establece y mantiene la integridad de las líneas base.                                |
| MA   | Desarrollar y mantener una capacidad de medición utilizada para dar soporte a las necesidades de información de gestión.  | <b>OE1. Alinear las actividades de medición y análisis:</b> los objetivos y actividades de medición están alineados con las necesidades y objetivos de información identificados.<br><b>OE2. Proporcionar resultados de medición:</b> se proporcionan los resultados de la medición, que abordan las necesidades y objetivos de información identificados. |
| PMC  | Monitorear y controlar el progreso del proyecto de tal manera que se puedan realizar las acciones correctivas cuando el cumplimiento del proyecto se desvíe del plan. | <b>OE1. Monitorear el proyecto frente al plan:</b> el progreso real del proyecto y el desempeño son monitoreados contra el plan del proyecto.<br><b>OE2. Gestionar acciones correctivas:</b> las acciones correctivas se logran hasta el cierre cuando el desempeño o los resultados del proyecto se desvían significativamente del plan.                  |
| PP   | Establecer y mantener la planificación de las actividades del proyecto.   | <b>OE1. Establecer estimaciones:</b> se establecen y mantienen estimaciones de los parámetros de planificación.<br><b>OE2. Desarrollar un plan de proyecto:</b> se establece y mantiene un plan para la gestión del proyecto.<br><b>OE3. Obtener compromiso con el plan:</b> se establecen y mantienen compromisos con el plan del proyecto.               |
| PPQA | Garantizar al personal y a la administración el cumplimiento objetivo de los procesos y productos de trabajo.   | <b>OE1. Revisión de cumplimiento:</b> la adherencia del proceso realizado y los productos de trabajo se evalúa de acuerdo a las descripciones y procedimientos establecidos.<br><b>OE2. Seguimiento y comunicación:</b> los problemas de incumplimiento se rastrean, comunican objetivamente y se garantiza la resolución.                                 |
| REQM | Gestionar los requisitos del producto y su alineación con los planes de proyecto.   | <b>OE1. Gestionar requisitos:</b> se gestionan los requisitos y se identifican las inconsistencias con los planes del proyecto y los productos de trabajo.   |
| SAM  | Gestionar la adquisición de productos y servicios de los proveedores.   | <b>OE1. Establecer acuerdos con proveedores:</b> se establecen y mantienen acuerdos con los proveedores.<br><b>OE2. Satisfacer los acuerdos de proveedores:</b> estos acuerdos son satisfactorios para el proyecto y proveedor.  |

**Fuente:** Elaboración propia

La evaluación parte desde el nivel más detallado que son las subprácticas hasta obtener el porcentaje de cada área, en la tabla tres se presenta el sistema de evaluación. Previo a la evaluación, se identificarán las estrategias SNAIL, cada una será representada con un código, el mismo que estará conformado por una letra y un número consecutivo. En la tabla cuatro, se describe la letra de acuerdo a la fase del proceso en que interviene la estrategia y se exponen algunos ejemplos.

**Tabla 3** Sistema de evaluación en base a CMMI-DEV nivel 2

| Variables   | Fórmulas   |
|---|--|
| <p><b>PAP:</b> puntaje del área de proceso.<br/> <b>POE:</b> puntaje del objetivo específico.<br/> <b>PPE:</b> puntaje de la práctica específica.<br/> <b>NOE:</b> número de objetivos específicos del área de proceso.<br/> <b>NPE:</b> número de prácticas específicas del objetivo específico.<br/> <b>NSE:</b> número de subprácticas de la práctica específica.<br/> <b>NSNAIL:</b> número de subprácticas soportadas por las estrategias SNAIL.</p> | $PPE = \frac{NSNAIL}{NSE} (100)$ $POE = \frac{\sum(PPE)}{NPE}$ $PAP = \frac{\sum(POE)}{NOE}$ |

**Fuente:** Elaboración propia

**Tabla 4** Código de las estrategias SNAIL

| Letra | Fase del Proceso        | Ejemplo | Letra | Fase del Proceso  | Ejemplo |
|-------|-------------------------|---------|-------|-------------------|---------|
| A     | Característica de SNAIL | A1      | E     | Programación      | E5      |
| B     | Requisitos              | B2      | F     | Pruebas           | F6      |
| C     | Planificación           | C3      | G     | Clausura          | G7      |
| D     | Diseño                  | D4      | H     | Inbound Marketing | H8      |

**Fuente:** Elaboración propia

Se aplicará las estrategias SNAIL en un caso de estudio, en el cual, se desarrollará un software web para la gestión automática de un sistema de aireación en la producción camaronera.

### 2.3 RESULTADOS

Se identificaron las estrategias SNAIL conforme se había establecido, las mismas que se constatan en la sección de anexos, (Ver Anexo B). Para verificar el soporte o cumplimiento en relación a los criterios de la ingeniería web y las áreas de proceso CMMI-DEV nivel 2, se indica el código de la estrategia. En la siguiente tabla, se presenta la evaluación y análisis en base a los criterios de la ingeniería web:

**Tabla 5** Estrategias SNAIL en base a los criterios de la ingeniería web.

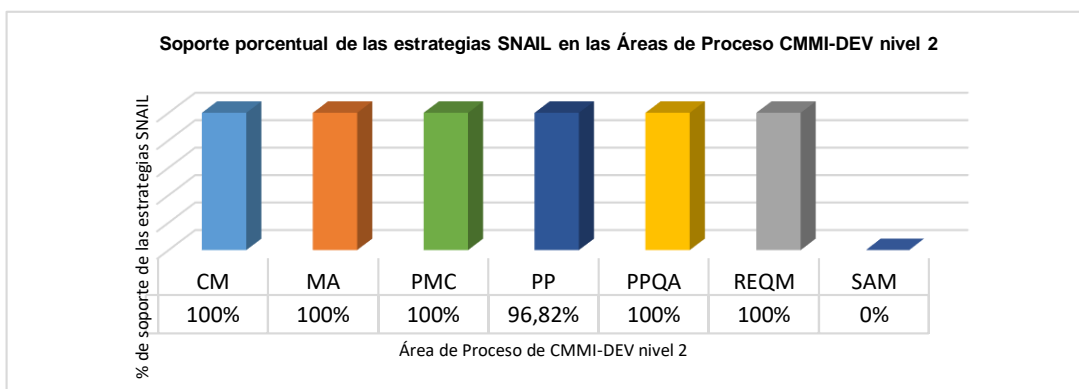
| Criterio | Código   | Análisis  |
|----------|--|---|
| AE       | A3, A6, A7, A9, B17, B19, C1-C5, D1, D4-D6, E1-E10, F1-F11, F14, G1-G7 | <p>Este criterio es soportado <b>SATISFACTORIAMENTE</b>.<br/>                     Si bien es cierto SNAIL no posee el nombre literal de las actividades estructurales, pero proporciona fases que cumplen objetivamente cada actividad.</p> <p>Área estructural de comunicación: una de las principales características de SNAIL es fomentar la interacción, colaboración y comunicación entre todos los interesados relevantes del proyecto, incluyendo al cliente como actor fundamental a lo largo del ciclo de desarrollo.</p> <p>Área estructural de planeación: la fase de planificación SNAIL, sugiere crear un marco o plan de trabajo denominado inicial, temporal o incremental, el cual debe actualizarse periódicamente conforme avanza el proyecto, involucrando estimación de recursos, tiempo, entregables, etc. En caso de cambios se debe reajustar la planificación.</p> <p>Área estructural de modelado: la fase de diseño SNAIL, abarca la creación de elementos y modelos como: base de datos, diccionario de datos, diseño conceptual, navegacional e interfaz abstracta; para asistir a los desarrolladores y clientes a entender los requerimientos del software.</p> |

| Criterio | Código | Análisis  |
|----------|--------|---|
| AE       |        | Área estructural de construcción: la fase de programación y pruebas SNAIL, permiten la creación de código y pruebas tanto individuales como integradas para detectar y solucionar errores.<br>Área estructural de implementación: la fase de clausura SNAIL, incluye la presentación de entregas incrementales, para que el cliente evalúe y proporcione retroalimentación del mismo.   |
| FP       | A1, A2 | Este criterio es soportado <b>SATISFACTORIAMENTE</b> .<br>Cabe recalcar que SNAIL es una metodología híbrida basada en metodologías ágiles (XP y SCRUM), por tanto, adopta estrategias para garantizar flexibilidad a los cambios que son un hecho en los proyectos de software web. Su flujo de proceso es un modelo en espiral, es decir iterativo.   |
| ED       | D4, D6 | Este criterio es soportado <b>SATISFACTORIAMENTE</b> .<br>Es claro que SNAIL no promueve la realización de los 6 elementos de diseño que sugiere la ingeniería web (interfaz, estética o diseño gráfico, contenido, arquitectónico, navegación y componentes), sin embargo cumple con la alternativa que es implementar métodos para diseño de hipermedia orientado a objetos, por lo que integra estrategias de la metodología de diseño OOHDM, la misma que utiliza modelos especializados como diseño conceptual, navegacional y diseño de interfaz abstracta, este es un punto relevante para el desarrollo de software web. Además, plantea la realización de base de datos a partir de los requisitos dividido en 3 diseños: conceptual, lógico y físico. |

**Fuente:** Elaboración propia

Las estrategias SNAIL soportan todos los criterios realizados por la ingeniería web, en este sentido es una metodología ideal para el desarrollo de proyectos de software web. Respecto a CMMI-DEV nivel 2, la evaluación en cada una de las áreas se verifica en la sección de anexos, (Ver Anexo C). En la ilustración uno, se presenta el resumen del soporte porcentual de las estrategias SNAIL por área de proceso:

#### Ilustración 1 Soporte de las estrategias SNAIL en CMMI-DEV nivel 2



**Fuente:** Elaboración propia

A continuación, el análisis de las estrategias SNAIL en cada una de las áreas:

**Tabla 6** Análisis de las estrategias SNAIL en base a CMMI-DEV nivel 2

| Área | Análisis  |
|------|---|
| CM   | <b>Está área es soportada en el 100%.</b><br><b>OE1. Establecer líneas base (100%):</b> el cumplimiento de este objetivo empieza con el planteamiento de los elementos del proyecto. Inicialmente en la fase de requisitos, se realiza el modelado de negocio para analizar la organización (cliente), a través del |

| Área | Análisis  |
|------|---|
| CM   | <p>reconocimiento de sus actividades, roles e información; se hace el estudio de factibilidad para estimar el tiempo, costos, beneficios, visión general del software, recursos económicos, técnicos, operativos, identificación de actores, asignación de responsabilidades, plan de tareas, plan alternativo y aprobación del proyecto; se ejecuta un proceso inicial de ingeniería de requisitos basado en la recolección, análisis, documentación y verificación de los mismos (aplicando técnicas y herramientas pertinentes). Luego en la fase de planificación, se crea el plan iterativo, el cual debe actualizarse en el caso de existir cambios ajustando tiempo y presupuesto; se establece en reunión con los actores relevantes, el cronograma de entregas, el mismo que se estima en base a las historias de usuario seleccionadas y ordenadas a prioridad del cliente. Se lleva un plan de contingencia y se conserva el control de versiones, lo cual permite la gestión del sistema de configuración del proyecto frente a posibles riesgos. Se indica los entregables de cada fase que darán apertura a la siguiente, entre estos: especificación de requisitos, casos de uso, historias de usuario, diccionario de datos, base de datos, diseños, pruebas, etc.</p> <p><b>OE2. Seguimiento y control de cambios (100%):</b> al ser un proceso iterativo, SNAIL acepta nuevos requisitos o modificaciones en cada iteración, los cuales son tratados en reunión con los actores indicados y si el caso lo amerita se realiza el ajuste de tiempo, presupuesto, recursos, cronograma de entregas, etc.</p> <p><b>OE3. Establecer Integridad (100%):</b> SNAIL enfoca la necesidad de mantener un archivo de cambios y elementos de configuración para mantenimiento y consecución de los requisitos del producto. Periódicamente se registra e informa los avances y problemas presentados para objetividad del plan y producto trazado.</p>                       |
| MA   | <p><b>Está área es soportada en el 100%.</b></p> <p><b>OE1. Alinear las actividades de medición y análisis (100%):</b> el soporte a este objetivo inicia con la identificación de los elementos a medirse, los mismos que se indicaron en el objetivo específico 1 (OE1) del área de Gestión de Configuración (CM). Para medir estos, dependerá de los aspectos relacionados a los mismos; en el caso de los recursos técnicos y operativos, del tamaño y dificultad del software previamente analizado; el tiempo de la especificación de requisitos, de la duración que conlleve las reuniones, entrevistas, disponibilidad de los interesados, etc; la estimación de entregas al cliente, de la complejidad de las historias de usuario consideradas en un rango de 1 a 3 puntos; la velocidad del equipo, de la experiencia de trabajo del mismo; el costo de desarrollo, del modelo utilizado sea este por estimación en líneas de código fuente (SLOC), no-SLOC o puntos de caso de uso.</p> <p><b>OE2. Proporcionar resultados de medición (100%):</b> una vez obtenidos los resultados de medición, son analizados, registrados y compartidos con los interesados para seguimiento, evaluación y control tanto del proyecto como del producto.</p>  |
| PMC  | <p><b>Está área es soportada en el 100%.</b></p> <p><b>OE1. Monitorear el proyecto frente al plan (100%):</b> SNAIL sugiere que el proyecto cuente con un plan de contingencia, lo cual permita direccionar las actividades del proceso en caso de desvíos significativos. Se lleva a cabo revisiones del progreso mediante la monitorización y evaluación constante de los parámetros de planificación con los actores responsables incluido el cliente. Se realiza reuniones con los actores relevantes dependiendo los puntos a tratarse, cada aspecto es aprobado con el acuerdo y compromiso de los mismos. En la existencia de cambios, persiste la información anterior para gestión de datos. Se presentan entregables al cliente, con la calidad y estado final conforme el cronograma. La evaluación iterativa del proyecto, permite evaluar el cumplimiento con el plan haciendo uso del estado actual del proyecto (identificación de retrasos, número de problemas, requisitos que se deben arrastrar para la siguiente iteración, etc), lista de historias de usuario que se deben completar, lista de riesgos, lista de cambios que se deben implementar al producto.</p> <p><b>OE2. Gestionar acciones correctivas (100%):</b> en la finalización de cada iteración se hace un análisis completo de lo que se cumplió o no, se habla del cumplimiento de los cambios de alcance aprobados durante el curso del proyecto, actividades de control de calidad, performance de ejecución de las actividades del programa, performance de ejecución del presupuesto, impacto sobre el servicio, evolución de supuestos riesgos, acuerdos para el soporte y evolución, reconciliación del presupuesto del proyecto y por último expresar las lecciones aprendidas para proceder a resolver los problemas con acciones correctivas y responsables encargados. Las acciones correctivas son monitorizadas en la siguiente reunión o iteración hasta su cierre completo.</p> |
| PP   | <p><b>Está área es soportada en el 96,82%.</b></p> <p><b>OE1. Establecer estimaciones (100%):</b> En la fase de requisitos se determina el alcance mediante el estudio del entorno de la organización (cliente), factibilidad del proyecto, factibilidad del software e ingeniería de requisitos. Luego en la fase de planificación de manera más detallada en cada iteración, se realiza un plan temporal donde se crea el cronograma de entregas en base a las historias de usuario priorizadas por el cliente y</p>  |



| Área | Análisis   |
|------|--|
| PP   | <p>tiempo pronosticado por los desarrolladores, el punto es la medida que equivale a una semana y cada historia de usuario representa 1 a 3 puntos. Al inicio de cada iteración se considera la velocidad del equipo y el costo de desarrollo se estima a través de 3 modelos: SLOC, no-SLOC o en puntos de caso de uso.</p> <p><b>OE2. Desarrollar un plan de proyecto (97,14%):</b> este objetivo es cumplido en la fase de requisitos y planificación, sin embargo, CMMI-DEV plantea la necesidad de contar con un espacio o directorio público para que los interesados puedan acceder a los datos gestionados del proyecto, por el contrario, SNAIL no menciona dicha estrategia, quedando a criterio del desarrollador implementarla o no. Es importante recordar que cada iteración culminada, permite obtener experiencia en las distintas actividades realizadas, tanto para el cliente, grupo de programadores, gerentes, directores y todos los actores del proyecto.</p> <p><b>OE3. Obtener compromiso con el plan (93,33%):</b> SNAIL promueve un entorno el cual fomenta la participación y comunicación de todos los interesados del proyecto (gerentes, directores, equipo de desarrollo, clientes) considerando que el cliente es un actor clave en todo el proceso de desarrollo. Todos los actores participan y son responsables de la configuración y ejecución de las tareas o actividades asignadas. En el caso de existir cambios en los requisitos o historias de usuario, se debe realizar una nueva reunión con los actores con el fin de evaluar el plan de entregas, estimar, ajustar el tiempo y costo. En la finalización de cada iteración o fase se debe dar a conocer un resumen, lo que involucra el expediente de las cosas que resultaron bien y mal, fortalezas, debilidades, procesos de gestión y acciones necesarias para culminar la fase o iteración. No obstante, SNAIL no especifica la documentación que se debe realizar para legalizar de manera jurídica los compromisos que tiene el cliente con los desarrolladores o viceversa.</p> |
| PPQA | <p><b>Está área es soportada en el 100%.</b></p> <p><b>OE1. Revisión de cumplimiento (100%):</b> La fase de clausura aplica a cada iteración, fase o proyecto en general, en todas incluye la aceptación de las actividades y entregables realizados para aprobación del cierre. En el caso de reprobación se deben realizar los debidos ajustes con las acciones correctivas garantizando así la objetividad del proceso y producto.</p> <p><b>OE2. Seguimiento y comunicación (100%):</b> Cualquier resultado obtenido de las pruebas de aceptación debe ser difundido en todos los equipos de desarrollo y corregir errores rápidamente si el caso lo requiere. En el cierre de cada fase o iteración se lista y transfiere los elementos con los detalles faltantes o completos. Se debe informar los cambios con la respectiva documentación.</p>   |
| REQM | <p><b>Está área es soportada en el 100%.</b></p> <p><b>OE1. Gestionar requisitos (100%):</b> SNAIL recomienda disponer de requisitos refinados y registrados que, al momento de construir el software, los programadores se centren en las correctas peticiones del cliente evitando sobrecostos, mala calidad, retrasos, cansancio en los miembros de equipo y por último clientes insatisfechos. Para ello, dispone un proceso de ingeniería de requisitos; la recolección hace uso de entrevistas (cerradas y abiertas) y casos de uso para representar los requisitos funcionales; para el análisis se plantean dos prácticas, la primera denominada JAD (Joint Application Development) la cual consiste en reuniones con el fin de definir los requerimientos válidos, eliminado aquellos que sean conflictivos sin mayor relevancia, la segunda práctica es la priorización de requerimientos mediante modelos; la especificación es el registro de los requisitos obtenidos en un documento el cual es compartido con todos los participantes del proyecto, cabe recalcar que este documento será fundamental en las evaluaciones, seguimiento y control tanto del producto como del proceso (diseño, pruebas, verificación, validación e implementación). El ciclo de desarrollo SNAIL especifica en sí mismo la trazabilidad directa entre los requisitos y el producto.</p>   |
| SAM  | <p><b>Está área no es soportada por las estrategias SNAIL.</b></p> <p>Esta área se aplica en entornos que adquieren componentes comerciales de venta directa (COTS, Commercial Off-The-Shelf), por el contrario, SNAIL no menciona comprar o adquirir componentes de software a proveedores, sin embargo, podrían considerarse ya que agilizarían el desarrollo, pero es importante analizar previamente los pros y contras que traerían consigo.</p>  |

**Fuente:** Elaboración propia

Finalmente, se desarrolló satisfactoriamente el software web (SHIRMPSystem) para la gestión del sistema de aireación en la producción camaronera, para ello se empleó la metodología SNAIL, el lenguaje de programación PHP, framework Laravel, base de datos MySQL y el servidor web Apache, (Ver Anexo D, E, F).



### 3. CONCLUSIONES

El software web es una tecnología muy utilizada que cada vez toma más fuerza, por lo cual es de vital importancia que sus desarrolladores adopten una metodología sistematizada y especializada en este entorno, de tal manera que puedan guiar su trabajo para la construcción de un producto final, sin mayores riesgos y que satisfaga las necesidades del cliente.

Se identificaron las estrategias que propone la metodología SNAIL en base al modelo CMMI-DEV nivel 2, obteniendo un excelente resultado en las áreas de proceso: Gestión de Configuraciones (CM), Medición y Análisis (MA), Monitoreo y Control del Proyecto (PMC), Aseguramiento de la Calidad del Proceso y del Producto (PPQA), Gestión de los Requerimientos (REQM). Sin embargo, en el área de Planeamiento del Proyecto (PP) tiene un mínimo inconveniente ya que no sugiere acceso público de los datos gestionados a todos los actores del proyecto; en el área de Gestión de los Acuerdos con Proveedores (SAM) presenta deficiencias, ya que no adquiere componentes (COTS) para desarrollo de software. Estas pequeñas falencias deben tomarse en cuenta por el desarrollador quedando a su criterio implementar estrategias pertinentes a estas áreas considerando siempre las ventajas y desventajas que generan.

De acuerdo a la ingeniería web, SNAIL es una metodología que propone estrategias ideales dentro del contexto y su implementación apunta a solucionar los problemas que hasta el momento han significado la crisis en este tipo de sistemas informáticos. En términos generales se concluye que SNAIL es una metodología evaluada satisfactoriamente y que sus estrategias son óptimas para el desarrollo de software web.

## REFERENCIAS BIBLIOGRÁFICAS

- [1] I. NASS DE LEDO, «LOS CINCUENTA AÑOS DE INTERNET,» *Revista Venezolana de Oncología*, vol. 31, nº 3, Julio-Septiembre 2019.
- [2] V. Esterkin y C. Pons, «Evaluación de calidad en el desarrollo de software dirigido por modelos,» *Ingeniare. Revista chilena de ingeniería*, vol. 25, nº 3, pp. 449-463, 2017.
- [3] J. R. Molina Ríos, M. P. Zea Ordoñez, J. A. Honores Tapia y A. S. Gómez Moreno, «Analysis Methodologies Web Application Development,» *International Journal of Applied Engineering Research*, vol. 11, nº 16, pp. 9070-9078, 2016.
- [4] J. R. Molina Ríos, M. P. Zea Ordoñez, F. G. García Zerda, M. J. Contenido Segarra y C. E. Jumbo Parrales, «SNAIL a hybrid model for the management of agile web software development processes,» *International Journal of Engineering Research and Technology*, vol. 11, nº 7, pp. 1067-1083, 2018.
- [5] M. P. Zea Ordoñez, J. R. Molina Ríos y L. F. Vincés Sánchez, «Analysis of the Benefits of Management Hotel Software in the City of Machala,» *Internal Journal of Advanced Engineering Research and Science*, vol. 3, pp. 2349-6495, 2016.
- [6] R. Pitre Redondo, H. Moscote Almanza, R. Curiel Gómez, J. E. Archila Guio y N. Amaya López, «Acceso y uso de la web 2.0 en los ambientes educativos étnicos de Riohacha-La Guajira,» *Revista Lasallista de Investigación*, vol. 14, nº 1, pp. 126-132, Junio 2017.
- [7] M. Callejas Cuervo, A. C. Alarcón Aldana y A. M. Álvarez Carreño, «Modelos de calidad del software, un estado del arte,» *Ingeniería y Tecnología*, vol. 13, nº 1, pp. 236-250, Enero-Junio 2017.
- [8] J. R. Molina Ríos, N. M. Loja Mora, M. P. Zea Ordoñez y E. L. Loaiza Sojos, «Evaluación de los Frameworks en el Desarrollo de Aplicaciones Web con Python,» *Revista Latinoamericana de Ingeniería de Software*, vol. 4, nº 4, pp. 201-207, 2016.
- [9] J. R. Molina Ríos, M. P. Zea Ordoñez, M. J. Contenido Segarra y F. G. García Zerda, «Estado del arte: metodologías de desarrollo en aplicaciones web,» *3C Tecnología*, vol. 6, nº 3, pp. 54-71, Septiembre-Diciembre 2017.

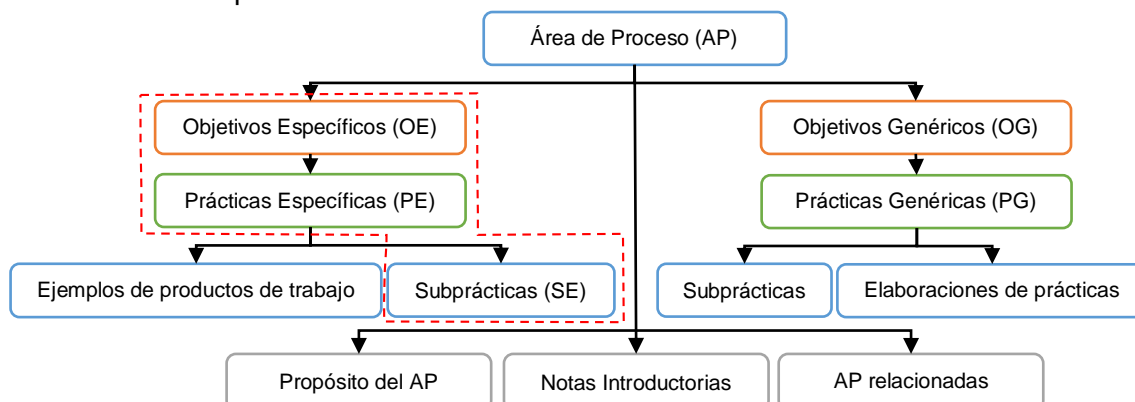
- [10] L. F. Londoño Rojas, V. Tabares Morales, M. Rosecler-Bez y N. D. Duque Mendez, «ANÁLISIS COMPARATIVO DE GUÍAS PARA EL DESARROLLO WEB ACCESIBLE,» *Ciencia e Ingeniería Neogranadina*, vol. 28, nº 1, pp. 101-115, Junio 2018.
- [11] J. L. Cantú Mata, F. Torres Castillo, S. Alcaraz Corona y F. Banda Muñoz, «CALIDAD, TIEMPO Y COSTO EN PROYECTOS DE DESARROLLO DE SOFTWARE,» *Interciencia*, vol. 43, nº 10, pp. 707-710, Octubre 2018.
- [12] E. García Sánchez, O. Vite Chávez, M. Á. Navarrate Sánchez, M. Á. García Sánchez y V. Torres Cosío, «Metodología para el desarrollo de software multimedia educativo MEDESME,» *CPU-e, Revista de Investigación Educativa*, vol. 1, nº 23, pp. 216-226, Julio-Diciembre 2016.
- [13] K. Curcio, R. Santana, S. Reinehr y A. Malucelli, «Usability in agile software development: A tertiary study,» *Computer Standards & Interfaces*, vol. 64, pp. 61-77, Mayo 2019.
- [14] J. Molina Ríos y M. Zea Ordoñez, «Metodologías de Desarrollo en Aplicaciones WEB,» *ARJÉ. Revista de Postgrado FaCE-UC*, vol. 11, nº 21, pp. 245-270, Julio-Diciembre 2017.
- [15] C. J. Madariaga Fernández, Y. Rivero-Peña y A. R. Leyva-Téllez, «Propuesta metodológica para desarrollo de software educativo en la Universidad de Holguín,» *Ciencias Holguín*, vol. 22, nº 4, pp. 1-17, Octubre-Diciembre 2016.
- [16] L. Kalb Roses, A. Windmöller y E. Almeida do Carmo, «FAVORABILITY CONDITIONS IN THE ADOPTION OF AGILE METHOD PRACTICES FOR SOFTWARE DEVELOPMENT IN A PUBLIC BANKING,» *JISTEM: Journal of Information Systems and Technology Management*, vol. 13, nº 3, pp. 439-458, Septiembre-Diciembre 2016.
- [17] Ö. Özcan-Top y O. Demirors, «Application of a Software Agility Assessment Model – AgilityMod in the Field,» *Computer Standards & Interfaces*, vol. 62, 2018.
- [18] B. Molina Montero, H. Vite Cevallos y J. Dávila Cuesta, «Metodologías ágiles frente a las tradicionales en el proceso de desarrollo de software,» *Espirales: Revista Multidisciplinaria de Investigación*, vol. 2, nº 17, 2018.

- [19] J. R. Molina Ríos, M. P. Zea Ordóñez, M. J. Contenido Segarra y F. G. García Zerda, «Comparación de metodologías en aplicaciones web,» *3C Tecnología*, vol. 7, nº 1, pp. 1-19, 2018.
- [20] I. Leiva Mundaca y M. Villalobos Abarca, «Método ágil híbrido para desarrollar en dispositivos móviles,» *Ingeniare. Revista Chilena de Ingeniería*, vol. 23, nº 3, pp. 473-488, 2015.
- [21] J. R. Molina Ríos, M. P. Zea Ordoñez, F. F. Redrován Castillo, N. M. Loja Mora, M. R. Valarezo Pardo y J. A. Honores Tapia, SNAIL, Una metodología híbrida para el desarrollo de aplicaciones web, Primera ed., Machala: 3Ciencias, 2018.
- [22] F. Selleri Silva, F. S. Furtado Soares, A. Lima Peres, I. Monteiro de Azevedo, A. P. Vasconcelos, F. Kenji Kamei y S. Romero de Lemos, «Using CMMI together with agile software development: A systematic review,» *Information and Software Technology*, vol. 58, pp. 20-43, Febrero 2015.
- [23] C. J. Torrecilla Salinas, J. Sedeño, M. J. Escalona y M. M., «Agile, Web Engineering and Capability Maturity Model Integration: A systematic literature review,» *Information and Software Technology*, vol. 71, nº 1, pp. 92-107, 2016.
- [24] T. G. Gonçalves, K. M. Oliveira y C. Kolski, «Identifying HCI approaches to support CMMI-DEV for interactive system development,» *Computer Standards & Interfaces*, vol. 58, pp. 53-86, Mayo 2018.
- [25] R. S. Pressman, Ingeniería del software. Un enfoque práctico, Séptima ed., México: McGRAW-HILL, 2010.
- [26] «Comparación de metodologías ágiles y procesos de desarrollo de software mediante un instrumento basado en CMMI,» *Scientia Et Technica*, vol. 21, nº 2, pp. 150-155, 2016.
- [27] Software Engineering Institute SEI, «CMMI-DEV, V1.3,» Carnegie Mellon University, Carnegie Mellon, 2010.

## ANEXOS

### ANEXO A: Esquema del Área de Proceso CMMI-DEV

**Ilustración 2** Esquema del Área de Proceso CMMI-DEV



**Fuente:** Elaboración propia

### ANEXO B: Identificación de las estrategias SNAIL

**Tabla 7** Estrategias SNAIL

| Código | Descripción de la estrategia   |
|--------|--|
| A1     | El modelo de sus fases es en espiral por lo cual su flujo de proceso es iterativo  |
| A2     | Compuesto por metodologías ágiles: XP y SCRUM  |
| A3     | Proceso basado en 7 fases: Requisitos, Planificación, Diseño, Programación, Pruebas, Clausura e Inbound marketing, las mismas que cubren las 5 actividades estructurales propuestas por la ingeniería web.   |
| A4     | Se realizan actividades en ciclos cortos denominados iteraciones, al culminar cada iteración se realiza una presentación de entregables o producto funcional.  |
| A5     | Cada iteración permite obtener experiencia en las distintas actividades realizadas, tanto para el cliente, grupo de programadores, gerentes, directores es decir todos los interesados (Stakeholders).   |
| A6     | SNAIL promueve un entorno el cual fomenta la participación y comunicación de todos los interesados del proyecto (gerentes, directores, equipo de desarrollo, clientes) y considera que el cliente es un actor clave en todo el proceso de desarrollo. Todos los actores participan de la configuración del proyecto donde cada uno tiene es responsable de las tareas o actividades asignadas. |
| A7     | SNAIL realiza reuniones con los actores del proyecto dependiendo los puntos a tratarse y cada aspecto es aprobado con el acuerdo y compromiso de dichos actores.   |
| A8     | El producto final o entregables son definidos mediante la estimación de tiempo, plasmado con fecha en el calendario del proyecto.  |
| A9     | SNAIL permite monitorear y evaluar constantemente los parámetros de planificación con la presencia de los responsables y sobretodo la participación del cliente. Las evaluaciones son debidamente registradas.   |
| B1     | Determinar el alcance del proyecto, mediante el entorno de la empresa (cliente), estudio de factibilidad, identificación de actores o usuarios. Y registrar en un documento de especificación de requisitos.   |
| B2     | SNAIL recomienda disponer de requisitos refinados y registrados que al momento de construir el software los programadores se centren en las correctas peticiones del cliente evitando sobrecostos, mala calidad, retrasos, cansancio en los miembros de equipo y por último clientes insatisfechos.  |
| B3     | Realizar un modelado de negocio con el fin de identificar los procesos de la organización (cliente). Obteniendo las actividades, roles, reglas e información en general, esto permitirá conocer los objetivos que debe cumplir el software.  |
| B4     | Diseñar diagramas de casos de uso para representar los procesos de negocio y los actores.  |
| B5     | Listar los actores que intervienen en el proceso de negocio y las actividades que realiza cada uno de ellos.   |
| B6     | Diseñar diagramas de flujo de actividades para representar en detalle el flujo de trabajo que incluye cada proceso de negocio.   |

| <b>Código</b> | <b>Descripción de la estrategia</b>  |
|---------------|--|
| <b>B7</b>     | Especificar las actividades representadas en diagrama de actividades, para esto SNAIL proporciona una plantilla como ejemplo.  |
| <b>B8</b>     | Realizar un estudio de factibilidad con el objetivo de estimar la duración y tamaño del proyecto, conocer costos y beneficios, determinar la factibilidad del sistema, estimación de recursos, personas que intervienen en el proyecto, planificación de alternativas, planificación de actividades y aprobación del proyecto.   |
| <b>B9</b>     | Describir de forma general al sistema (estudio de factibilidad).   |
| <b>B10</b>    | Definir en cantidad y costo los recursos del proyecto: personal, recursos técnicos y materiales (estudio de factibilidad).   |
| <b>B11</b>    | Identificar los usuarios del sistema (estudio de factibilidad).  |
| <b>B12</b>    | Identificar los beneficios esperados del proyecto (estudio de factibilidad).   |
| <b>B13</b>    | Análisis de factibilidad económica, técnica y operativa (estudio de factibilidad).   |
| <b>B14</b>    | Determinar el impacto que trae el software a la organización (cliente) en lo que respecta a su planeación estratégica (estudio de factibilidad).   |
| <b>B15</b>    | Cronograma de actividades (estudio de factibilidad).   |
| <b>B16</b>    | SNAIL proporciona un proceso de ingeniería de requisitos basado en la recolección, análisis, especificación o documentación y verificación de requisitos.  |
| <b>B17</b>    | Para la recolección de requisitos SNAIL sugiere hacer uso de entrevistas (cerradas y abiertas) y casos de uso para representar los requisitos funcionales del sistema.   |
| <b>B18</b>    | Cabe recalcar que SNAIL posee un modelo de flujo iterativo por lo tanto sería posible aceptar modificaciones en los requisitos con el avance del proyecto, dichos cambios son documentados.  |
| <b>B19</b>    | Para el análisis de requisitos SNAIL plantea dos prácticas, la primera denominada JAD (Joint Application Development) la cual consiste en reuniones donde se integra el cliente, directivos y desarrolladores con el fin de definir los requerimientos que serán válidos en el desarrollo del software, eliminando aquellos que sean conflictivos sin mayor relevancia. La segunda práctica es la priorización de requerimientos mediante modelos que permite dar mayor valor a los requerimientos evaluándolos de acuerdo a las características de un buen requerimiento. |
| <b>B20</b>    | La especificación de requisitos es el registro de los mismos en un documento el cual es compartido con todos los participantes del proyecto, cabe recalcar que este documento será fundamental en las evaluaciones, seguimiento y control tanto del producto como del proceso (diseño, pruebas, verificación, validación e implementación).  |
| <b>B21</b>    | Se debe verificar internamente (es decir con los miembros del equipo de proyecto) y externamente (con el cliente) los requisitos especificados en el documento para garantizar consistencia y exactitud en los requisitos.   |
| <b>B22</b>    | SNAIL menciona algunas herramientas y técnicas extras que sirven de apoyo en las fases (recolección, análisis, especificación y validación) de la ingeniería de requerimientos.  |
| <b>B23</b>    | SNAIL sugiere software para la administración de requerimientos entre estos: DOORS, RTM Workshop, Caliber -RM, RequisitePro.   |
| <b>C1</b>     | SNAIL sugiere crear un marco de trabajo temporal o inicial el cual debe actualizarse periódicamente conforme avanza el proyecto, debido a que su flujo de proceso es iterativo. Este marco de trabajo se trata de la planificación del proyecto lo cual involucra estimación de recursos, tiempo, entregables, etc. Cabe recalcar que estas estimación deben considerar la presentación posibles casos negativos es decir crear un plan de contingencia.   |
| <b>C2</b>     | Se reemplaza la documentación funcional y los casos de uso por historias de usuario las cuales son escritas por el cliente con un detalle mínimo para estimación de tiempo más acertado y menos riesgoso.  |
| <b>C3</b>     | Mediante una reunión con todos los actores del proyecto se crea el calendario de entregas. Una entrega es un conjunto de historias de usuario las mismas que son agrupadas por el cliente. El orden de entrega se realiza en base a la prioridad del cliente.  |
| <b>C4</b>     | El cronograma de entregas se programa en base a la estimación del tiempo de desarrollo que realiza el equipo de programadores.   |
| <b>C5</b>     | En el caso de existir cambios en los requisitos o historias de usuario, se debe realizar una nueva reunión con todos los actores del proyecto con el fin de evaluar el plan de entregas, estimar y ajustar el tiempo y costo.  |
| <b>C6</b>     | Es importante mencionar que con la existencia de cambios persiste la información anterior. Como por ejemplo cambios en las historias de usuario lo que conlleva a ajustar el plan de entregas, por tanto se conserva las historias de usuario y plan de entregas anteriores (es decir control de versiones).   |
| <b>C7</b>     | Las estimaciones de entregables se basan en un conjunto de historias de usuario y el tiempo asociado a ellos, los programadores usan como medida el punto lo que equivale a una semana y cada historia de usuario se estima entre 1 a 3 puntos.  |
| <b>C8</b>     | Los atributos del producto se estiman en base a las mismas historias de usuario en el caso de tratarse de historias complicadas estas son divididas o si fuere sencilla se complementa con otra.   |

| Código | Descripción de la estrategia   |
|--------|--|
| C9     | Una correcta estimación de costo permitirá conocer el esfuerzo y tiempo invertido en la construcción de un proyecto de software. SNAIL plantea 3 modelos de estimación de costos los cuales son: modelo basado en líneas de código fuente (SLOC), modelo no-SLOC y modelo basado en puntos de casos de uso.  |
| C10    | SNAIL sugiere calcular la velocidad del proyecto es decir la capacidad del equipo para desarrollar un trabajo en una determinada iteración. Esta velocidad permite planificar el proyecto en base al tiempo o al alcance.  |
| C11    | Un punto importante que plantea SNAIL es el escenario de Benchmarking, lo cual permite mejorar procesos, servicios o productos basándose en la comparación de otras empresas líderes o de la misma empresa. En SNAIL aplicando Benchmarking interno permitirá aprender de las prácticas o procesos realizados en cada iteración o en cada proyecto lo cual rescata los aspectos positivos y rechazaría los negativos por posibles problemas presentados. |
| D1     | El diseño del software se reestructura y organiza en elementos que puedan ser desarrollados de manera individual, aprovechando los beneficios del desarrollo en equipo.  |
| D2     | Para realizar el diseño en SNAIL solo se considera las historias de usuario que conforman los entregables en la iteración propuesta, puesto que la naturaleza del software y por ende del proyecto es cambiante y realizar un diseño complejo en las fases iniciales podría ser una pérdida de tiempo.   |
| D3     | El diseño al igual que las demás fases de SNAIL consiste en una constante actividad de corrección y mejoramiento con el avance del proyecto, todos los cambios realizados se registrar para el debido seguimiento y control en la evaluación del producto final.   |
| D4     | El diseño de la base de datos parte de los requisitos obtenidos en este caso de las historias de usuario que conforman el entregable, a la elaboración de 3 diseños los cuales son: conceptual, lógico y físico.   |
| D5     | SNAIL recomienda realizar el diccionario de datos para describir todos los datos que integran el software para que los analistas y usuarios puedan identificarlos de manera clara. Cabe recalcar la importancia del diccionario de datos ya que permite estar al tanto de los detalles de un software extenso, permite una mejor comunicación entre los programadores o desarrolladores y permite localizar de manera fácil errores en el software.      |
| D6     | SNAIL integra estrategias de la metodología de diseño OOHDM, la misma que utiliza modelos especializados como diseño conceptual, navegacional y diseño de interfaz abstracta, cabe recalcar que este tipo de diseños son propuestas por la ingeniería web que otras metodologías no toman en cuenta y por tanto SNAIL toma mayor relevancia para el desarrollo de software web.  |
| E1     | Esta fase consiste en realizar el código fuente para creación del software, mediante el uso de prototipos, pruebas, detección y corrección de errores. Se identifican los componentes que se reutilizarían para ganar tiempo en el proceso de programación.  |
| E2     | Para la codificación SNAIL propone el uso de repositorios donde la pareja de programadores pueda subir el código corregido en base a pruebas realizadas.   |
| E3     | SNAIL propone que cualquiera de los programadores puede modificar o ampliar código de una clase o método a cargo de otro programador si el caso lo amerita.  |
| E4     | SNAIL sugiere la recodificación cada vez que sea necesario con el objetivo de mantener un código simple y que cumpla con la funcionalidad deseada.   |
| E5     | La codificación en SNAIL se realiza en pares de programadores, ya que de esta manera se reduce de manera significativa los errores, puesto que se descubre en el tiempo de codificación, código más simple y mejores diseños, resolución rápida de problemas, los miembros de desarrollo aprenden más del software y cada parte del código en detalle.   |
| E6     | Al igual que en otras fases, SNAIL requiere de la disponibilidad del cliente en la fase de codificación, así se cumple con un software cuyas funcionalidades sean las requeridas por el cliente.   |
| E7     | SNAIL propone el uso de estándares para la codificación como es el caso del diccionario de datos, esto hace que el código implementado sea comprensible por todo el equipo.  |
| E8     | SNAIL promueve que los módulos sean evaluados de manera individual es decir que la codificación sea sometida a pruebas unitarias antes de su publicación en el repositorio, en el caso de detectar errores se deben corregir, esto garantiza que funcione el código de manera colectiva.   |
| E9     | SNAIL sugiere realizar un diagrama de estructuras basado en jerarquías el cual represente la interconexión de los módulos del software.  |
| E10    | SNAIL plantea que el caso de existir un retraso muy relevante en la codificación se debe renegociar el plan de entregas mediante una reunión de planificación con todos los actores del proyecto.  |
| F1     | Los módulos programados deben integrarse para complementar el software y se pone a prueba su funcionalidad y requisitos sugeridos por el cliente.  |



| Código | Descripción de la estrategia   |
|--------|--|
| F2     | Se realizan pruebas unitarias también llamadas pruebas de componentes, módulos o clases son realizadas por parte de los programadores, se realizan varias incluyendo el ingreso de datos erróneos e inesperados.   |
| F3     | Se realizan pruebas integrando todos los componentes, estas pruebas son realizadas por parte de los programadores.   |
| F4     | Se realizan pruebas del software en un ambiente pre-producción estas pruebas son realizadas por programadores ajenos a quien desarrollo la codificación o terceros, se verifica la funcionalidad completa del software en base a las documentación de los requisitos registrados, requisitos funcionales y no funcionales, para realizar estas pruebas el equipo se basa en entregables como: requerimientos iniciales, casos de uso, historias de usuario, etc.   |
| F5     | En el caso de detectar errores en las pruebas del software se corrigen inmediatamente y al mismo tiempo se realizan pruebas para verificar que el error ha sido solucionado, lo que significa control de las acciones correctivas.   |
| F6     | Una proceso de pruebas que recomienda SNAIL es la que utiliza la metodología RUP, las cual consiste en 5 etapas: planeación, diseño, implementación de pruebas, evaluación de criterios de salida y cierre del proceso.  |
| F7     | Los clientes deben verificar el resultado positivo o negativo de las pruebas en base a las historias de usuario o entregables que fueron implementados.  |
| F8     | En el caso de que el cliente encuentre errores debe informar a los programadores para su resolución  |
| F9     | Una historias de usuario o entregable no se considera culminada mientras se encuentre en las pruebas de verificación cuya responsabilidad de aprobación recae en el cliente  |
| F10    | Cualquier resultado obtenidos de las pruebas de aceptación debe ser difundido en todos los equipos de desarrollo y corregir errores rápidamente si el caso lo requiere.  |
| F11    | Periódicamente se debe registrar e informar sobre los avances y posibles problemas presentados en el proyecto, se debe realizar un seguimiento de los recursos utilizados y que el plan se está ejecutando en los tiempos estimados.   |
| F12    | Periódicamente se debe hacer un análisis del impacto directo que causan los cambios en el proyecto.  |
| F13    | El proyecto debe contar con una planificación con cierto margen de maniobra o plan de contingencia, lo cual permita direccionar las actividades o procesos del proceso sin sobrecostos u otros problemas.  |
| F14    | Proponer al cliente cambios para mejorar el producto respecto a lo que ofrecen posibles competidores.  |
| G1     | La fase de clausura aplica tanto a cada iteración, al proyecto en general y cada fase, todas incluye la aceptación de las actividades y entregables realizados para aprobación al cierre. En el caso de reprobación se deben realizar los debidos ajustes con las acciones correctivas.  |
| G2     | Esta fase es utilizada para informar los éxitos y fracasos del proyecto, lecciones aprendidas, evaluación de la ejecución real en base a la planificación, todo esto debe ser considerado para mejorar en las próximas iteraciones o proyectos.  |
| G3     | En el cierre de cada fase o iteración se lista y transfiere los elementos con los detalles faltantes o completos. Se debe informar los cambios con la respectiva documentación.  |
| G4     | En la finalización de cada iteración o proyecto se debe realizar una reunión de revisión con todos los actores del proyecto para dar a conocer el resumen del proyecto, lo que involucra la documentación de las cosas que resultaron bien y mal, fortalezas, debilidades, procesos de gestión y acciones necesarias para culminar el proyecto o iteración.  |
| G5     | Presentar los entregables al cliente, con la calidad y estado final en base a la planificación del calendario de entregas.   |
| G6     | La fase de clausura lo que respecta a presentación de entregables se hace un análisis completo de lo que se cumplió o no, se habla del cumplimiento de los cambios de alcance aprobados durante el curso del proyecto, actividades de control de calidad, performance de ejecución de las actividades del programa, performance de ejecución del presupuesto, impacto sobre el servicio, evolución de supuestos y riesgos, acuerdos para el soporte y evolución, reconciliación del presupuesto del proyecto y por último expresar las lecciones aprendidas para proceder a resolver los problemas con acciones correctivas y responsables encargados. |
| G7     | La evaluación iterativa del proyecto, permite evaluar el cumplimiento con el plan del proyecto haciendo uso del estado actual del proyecto (identificación de retrasos, número de problemas, requisitos que se deben arrastrar para la siguiente iteración), lista de historias de usuario que se deben completar, lista de riesgos, lista de cambios que se deben implementar al producto.  |
| H1     | La fase de Inbound marketing consiste en difundir el software a través de blogs, buscadores, redes sociales, etc.  |

Fuente: Elaboración propia

**ANEXO C:** Evaluación de las estrategias SNAIL en las áreas de proceso CMMI-DEV nivel 2

**Tabla 8** Evaluación de las estrategias SNAIL en el área de proceso – CM

| <b>Gestión de Configuraciones (CM)</b>  |  |            |               | <b>100%</b> |
|---|--|------------|---------------|-------------|
| <b>OE1. Establecer líneas base</b>  |  |            |               | 100%        |
| <b>OE2. Seguimiento y control de cambios</b>  |  |            |               | 100%        |
| <b>OE3. Establecer integridad</b>   |  |            |               | 100%        |
| <b>Prácticas Específicas (PE)</b>   | <b>Estrategias</b>                         | <b>NSE</b> | <b>NSNAIL</b> | <b>%</b>    |
| PE1.1 Identificar los elementos de configuración.   | A6, A8, B5, B8-B21, C1-C4, C8              | 6          | 6             | 100%        |
| PE1.2 Establecer y mantener un sistema de gestión de configuración.   | C6, D3, F12, F13                           | 9          | 9             | 100%        |
| PE1.3 Crear o liberar las líneas base.  | B20, B21, C2-C4, D4-D6, E8, E9, F7, G1, G3 | 4          | 4             | 100%        |
| PE2.1 Seguir las peticiones de cambio para los elementos de configuración.                                  | B18, C5, C6                                | 5          | 5             | 100%        |
| PE2.2 Controlar los cambios a los elementos de configuración.   | C6, D3, F12, F13                           | 5          | 5             | 100%        |
| PE3.1 Establecer y mantener los registros que describen los elementos de configuración.                     | C6, D3, F12, F13                           | 6          | 6             | 100%        |
| PE3.2 Realizar auditorías de configuración para mantener la integridad de las líneas base de configuración. | F11-F13                                    | 6          | 6             | 100%        |

**Fuente:** Elaboración propia

**Tabla 9** Evaluación de las estrategias SNAIL en el área de proceso – MA

| <b>Medición y Análisis (MA)</b>   |  |            |               | <b>100%</b> |
|---|--|------------|---------------|-------------|
| <b>OE1. Alinear las actividades de medición y análisis</b>                  |  |            |               | 100%        |
| <b>OE2. Proporcionar resultados de medición</b>                             |  |            |               | 100%        |
| <b>Prácticas Específicas (PE)</b>   | <b>Estrategias</b>                       | <b>NSE</b> | <b>NSNAIL</b> | <b>%</b>    |
| PE1.1 Establecer objetivos de medida.                                       | C1, C2, C4, C5, C7-C10                   | 5          | 5             | 100%        |
| PE1.2 Especificar medidas.  | C1, C2, C4, C5, C7-C10                   | 5          | 5             | 100%        |
| PE1.3 Especificar procedimientos de recopilación y almacenamiento de datos. | A8, B2, B16-B23, E2                      | 7          | 7             | 100%        |
| PE1.4 Especificar procedimientos de análisis.                               | C1, C2, C4, C5, C7-C10                   | 6          | 6             | 100%        |
| PE2.1 Obtener los datos de la medición.                                     | C1, C2, C4, C5, C7-C10                   | 3          | 3             | 100%        |
| PE2.2 Analizar los datos de la medición.                                    | C1, C2, C4, C5, C7-C10                   | 4          | 4             | 100%        |
| PE2.3 Almacenar datos y resultados.   | B20, B22, D3, D5, E2, E9, F11, G4        | 4          | 4             | 100%        |
| PE2.4 Comunicar los resultados.   | A7, C5, E10, F6, F7, F8, F10, F11, G1-G4 | 2          | 2             | 100%        |

**Fuente:** Elaboración propia

**Tabla 10** Evaluación de las estrategias SNAIL en el área de proceso – PMC

| <b>Monitoreo y Control del Proyecto (PMC)</b>                   |   |            |               |          | <b>100%</b> |
|---|---|------------|---------------|----------|-------------|
| <b>OE1. Monitorizar el proyecto frente al plan</b>              |   |            |               |          | 100%        |
| <b>OE2. Gestionar acciones correctivas</b>                      |   |            |               |          | 100%        |
| <b>Prácticas Específicas (PE)</b>                               | <b>Estrategias</b>  | <b>NSE</b> | <b>NSNAIL</b> | <b>%</b> |             |
| PE1.1 Monitorizar los parámetros de planificación del proyecto. | A5, A9, C5, C6, C11, D3, E10, F7, F8, F10-F12, G1, G2, G4, G6, G7 | 6          | 6             | 100%     |             |
| PE1.2 Monitorizar los compromisos.                              | A6, A7, C5, E6, F7, F11, G6, G7                                   | 3          | 3             | 100%     |             |
| PE1.3 Monitorizar riesgos.                                      | B1, B2, B8, C1, F8, F10-F13, G4, G6, G7                           | 3          | 3             | 100%     |             |
| PE1.4 Monitorizar la gestión de datos.                          | B22, C3, C6, D3, E9, F11, G1, G3                                  | 3          | 3             | 100%     |             |
| PE1.5 Monitorizar la participación de los involucrados.         | A6, B18, C5, D3, E6, F10  | 3          | 3             | 100%     |             |
| PE1.6 Llevar a cabo revisiones del progreso.                    | A9, B18, C5, C6, F9, F10, F11, G4-G7                              | 6          | 6             | 100%     |             |
| PE1.7 Llevar a cabo revisiones de hitos.                        | G5  | 5          | 5             | 100%     |             |
| PE2.1 Analizar problemas.                                       | F2-F5, F7, F10, F11, G4, G6, G7                                   | 2          | 2             | 100%     |             |
| PE2.2 Llevar a cabo medidas correctivas.                        | A7, C5, F10, G1, G4, G6   | 3          | 3             | 100%     |             |
| PE2.3 Gestionar las medidas correctivas.                        | A5, F5, G1, G2, G4, G7  | 3          | 3             | 100%     |             |

**Fuente:** Elaboración propia

**Tabla 11** Evaluación de las estrategias SNAIL en el área de proceso – PP

| <b>Planificación de Proyectos (PP)</b>   |  |            |               |          | <b>96,82%</b> |
|--|--|------------|---------------|----------|---------------|
| <b>OE1. Establecer estimadores</b>   |  |            |               |          | 100%          |
| <b>OE2. Desarrollar un plan de proyecto</b>                                      |  |            |               |          | 97,14%        |
| <b>OE3. Obtener compromiso con el plan</b>                                       |  |            |               |          | 93,33%        |
| <b>Prácticas Específicas (PE)</b>  | <b>Estrategias</b>                       | <b>NSE</b> | <b>NSNAIL</b> | <b>%</b> |               |
| PE1.1 Estimar el alcance del proyecto.   | B1, B2, B8, B16-B21, C1, C3, C4          | 4          | 4             | 100%     |               |
| PE1.2 Establecer una estimación del producto de trabajo y las tareas a realizar. | B8-B15, C2-C4, C7, C8                    | 3          | 3             | 100%     |               |
| PE1.3 Definir las fases del ciclo de vida del proyecto.                          | A3                                       | 1          | 1             | 100%     |               |
| PE1.4 Estimar el esfuerzo y el coste.  | A5, C3, C5, C6, C9, C10, G4, G6          | 3          | 3             | 100%     |               |
| PE2.1 Establecer el calendario y los costes del proyecto.                        | B8, C3-C5, C10, F11, F12, G1, G3, G4, G6 | 6          | 6             | 100%     |               |
| PE2.2 Identificar los riesgos.   | B2, B8, F12, F13, G4, G6, G7             | 4          | 4             | 100%     |               |
| PE2.3 Planificar la gestión de datos.  | A6, A7, B1, B18, B22, C3, C4, C6, E2     | 5          | 4             | 80%      |               |
| PE2.4 Planificar los recursos del proyecto.                                      | A6, B8, B21, C3, C5, F7                  | 5          | 5             | 100%     |               |
| PE2.5 Planificar las habilidades y conocimientos necesarios.                     | A5, B8, F13, G2                          | 4          | 4             | 100%     |               |
| PE2.6 Planificar la participación de los involucrados.                           | A6                                       | 1          | 1             | 100%     |               |
| PE2.7 Establecer el plan del proyecto.   | B8-B15, B17-B21, C1-C4                   | 1          | 1             | 100%     |               |
| PE3.1 Revisar los planes que puedan afectar al proyecto.                         | A7, B18, C2, C4-C6, C10, D2, G2, G4-G7   | 1          | 1             | 100%     |               |

|   |                     |   |   |      |
|---|---------------------|---|---|------|
| PE3.2 Poner de acuerdo el trabajo y los recursos compartidos. | B18, C5, F7, G5,G6  | 1 | 1 | 100% |
| PE3.3 Obtener compromiso de acuerdo para el plan de proyecto. | A6, B20, B21, C2-C5 | 5 | 4 | 80%  |

**Fuente:** Elaboración propia

**Tabla 12** Evaluación de las estrategias SNAIL en el área de proceso – PPQA

| <b>Planificación de Proyectos (PPQA)</b>           |   |            |               | <b>100%</b> |
|--|---|------------|---------------|-------------|
| <b>OE1. Revisión de cumplimiento</b>               |   |            |               | 100%        |
| <b>OE2. Seguimiento y comunicación</b>             |   |            |               | 100%        |
| <b>Prácticas Específicas (PE)</b>                  | <b>Estrategias</b>  | <b>NSE</b> | <b>NSNAIL</b> | <b>%</b>    |
| PE1.1 Evaluación objetiva de procesos.             | A5, A6, A9, B3-B21, C2-C6, D4-D6, E5, E6, F9, G1, G2      | 5          | 5             | 100%        |
| PE1.2 Evaluación objetiva de productos de trabajo. | A5, A6, A9, B3-B21, C2-C6, D4-D6, E5, E6, F9, G1, G2      | 6          | 6             | 100%        |
| PE2.1 Comunicar y resolver no conformidades.       | A7, C5, E3, E10, F5, F7, F8, F10, F11, G4, G1- G4, G6, G7 | 7          | 7             | 100%        |
| PE2.2 Establecer registros.                        | G1, G3  | 2          | 2             | 100%        |

**Fuente:** Elaboración propia

**Tabla 13** Evaluación de las estrategias SNAIL en el área de proceso – REQM

| <b>Gestión de los Requerimientos (REQM)</b>                                 |                                 |            |               | <b>100%</b> |
|---|---------------------------------|------------|---------------|-------------|
| <b>OE1. Gestionar requisitos</b>  |                                 |            |               | 100%        |
| <b>Prácticas Específicas (PE)</b>   | <b>Estrategias</b>              | <b>NSE</b> | <b>NSNAIL</b> | <b>%</b>    |
| PE1.1 Comprender los requisitos.  | B3-B7, B16, B17, B19, B20, B21  | 4          | 4             | 100%        |
| PE1.2 Obtener compromisos para los requisitos.                              | B1, C5                          | 2          | 2             | 100%        |
| PE1.3 Gestionar los cambios de requisitos.                                  | B18, C5, C6                     | 4          | 4             | 100%        |
| PE1.4 Mantener una trazabilidad bidireccional o de los requisitos.          | F4, F7-F10, G5                  | 3          | 3             | 100%        |
| PE1.5 Asegurar alineamiento entre el trabajo del proyecto y los requisitos. | A5, B2, C5, C11, E8, E10, G5-G7 | 4          | 4             | 100%        |

**Fuente:** Elaboración propia

**Tabla 14** Evaluación de las estrategias SNAIL en el área de proceso – SAM

| <b>Gestión de los Acuerdos con Proveedores (SAM)</b> |                    |            |               | <b>0%</b> |
|--|--------------------|------------|---------------|-----------|
| <b>OE1. Establecer acuerdos con proveedores</b>      |                    |            |               | 0%        |
| <b>OE2. Satisfacer los acuerdos con proveedores</b>  |                    |            |               | 0%        |
| <b>Prácticas Específicas (PE)</b>                    | <b>Estrategias</b> | <b>NSE</b> | <b>NSNAIL</b> | <b>%</b>  |
| PE1.1 Determinar el tipo de adquisición.             | NINGUNA            | 1          | 0             | 0%        |
| PE1.2 Seleccionar a los proveedores.                 | NINGUNA            | 6          | 0             | 0%        |
| PE1.3 Establecer acuerdos con proveedores.           | NINGUNA            | 7          | 0             | 0%        |

|  |         |   |   |    |
|--|---------|---|---|----|
| PE2.1 Ejecutar el acuerdo de proveedor.        | NINGUNA | 8 | 0 | 0% |
| PE2.2 Aceptar el producto adquirido.           | NINGUNA | 7 | 0 | 0% |
| PE2.3 Asegurar la transición de los productos. | NINGUNA | 3 | 0 | 0% |

Fuente: Elaboración propia

## ANEXO D: Requisitos funcionales del software web SHIRMPSystem

**Tabla 15** RF01 – Gestión de piscinas

|                                    |  |
|------------------------------------|--|
| <b>Identificador del requisito</b> | RF01   |
| <b>Nombre del requisito</b>        | Gestión de piscinas  |
| <b>Tipo</b>                        | Funcional  |
| <b>Descripción</b>                 | Permite al usuario registrar, editar, eliminar y listar las piscinas que mantiene la camaronera, los campos correspondientes a una piscina son: nombre, ubicación, número de hectáreas además se deberá asignar un aireador a dicha piscina. |
| <b>Actores</b>                     | Usuario  |
| <b>Prioridad</b>                   | Alta   |
| <b>Condición de éxito</b>          | La piscina fue registrada y gestionada con éxito   |
| <b>Condición de fracaso</b>        | Los campos no son válidos al gestionar las piscinas  |

Fuente: Elaboración propia

**Tabla 16** RF02 – Gestión de especies

|                                    |   |
|------------------------------------|---|
| <b>Identificador del requisito</b> | RF02  |
| <b>Nombre del requisito</b>        | Gestión de especies   |
| <b>Tipo</b>                        | Funcional   |
| <b>Descripción</b>                 | Permite al usuario registrar, editar, eliminar y listar las especies que mantiene la camaronera, los campos correspondientes a una especie son: nombre, descripción y las medidas máximas mínimas de los parámetros del agua (pH, Oxígeno Disuelto y Temperatura) sobre los que debe vivir dicha especie. |
| <b>Actores</b>                     | Usuario   |
| <b>Prioridad</b>                   | Alta  |
| <b>Condición de éxito</b>          | La especie fue registrada y gestionada con éxito  |
| <b>Condición de fracaso</b>        | Los campos no son válidos al gestionar las especies   |

Fuente: Elaboración propia

**Tabla 17** RF03 – Gestión de aireadores

|                                    |  |
|------------------------------------|--|
| <b>Identificador del requisito</b> | RF03   |
| <b>Nombre del requisito</b>        | Gestión de aireadores  |
| <b>Tipo</b>                        | Funcional  |
| <b>Descripción</b>                 | Permite al usuario registrar, editar, eliminar y listar el aireador asignado en la piscina, los campos correspondientes a un aireador son: nombre, dirección ip y mac. |
| <b>Actores</b>                     | Usuario  |
| <b>Prioridad</b>                   | Alta   |
| <b>Condición de éxito</b>          | El aireador fue registrado y gestionado con éxito  |
| <b>Condición de fracaso</b>        | Los campos no son válidos al gestionar los aireadores  |

Fuente: Elaboración propia

**Tabla 18** RF04 – Gestión de corridas

|                                    |   |
|------------------------------------|---|
| <b>Identificador del requisito</b> | RF04  |
| <b>Nombre del requisito</b>        | Gestión de corridas   |
| <b>Tipo</b>                        | Funcional   |
| <b>Descripción</b>                 | Permiten al usuario registrar, editar, eliminar y listar las corridas realizadas por la camaronera. Los campos correspondientes a una corrida son: nombre, descripción, periodo y semestre. |
| <b>Actores</b>                     | Usuario   |
| <b>Prioridad</b>                   | Alta  |
| <b>Condición de éxito</b>          | La corrida fue registrada y gestionada con éxito  |
| <b>Condición de fracaso</b>        | Los campos no son válidos al gestionar las corridas   |

**Fuente:** Elaboración propia

**Tabla 19** RF05 – Gestión de siembras

|                                    |   |
|------------------------------------|---|
| <b>Identificador del requisito</b> | RF05  |
| <b>Nombre del requisito</b>        | Gestión de siembras   |
| <b>Tipo</b>                        | Funcional   |
| <b>Descripción</b>                 | Permiten al usuario registrar, editar, eliminar y listar las siembras realizadas por la camaronera. Los campos de una siembra son: nombre o descripción, observación, corrida, piscina, especie, número de larvas, fecha y se debe especificar si dicha siembra ha sido o no cosechada. |
| <b>Actores</b>                     | Usuario   |
| <b>Prioridad</b>                   | Alta  |
| <b>Condición de éxito</b>          | La siembra fue registrada y gestionada con éxito  |
| <b>Condición de fracaso</b>        | Los campos no son válidos al gestionar las siembras   |

**Fuente:** Elaboración propia

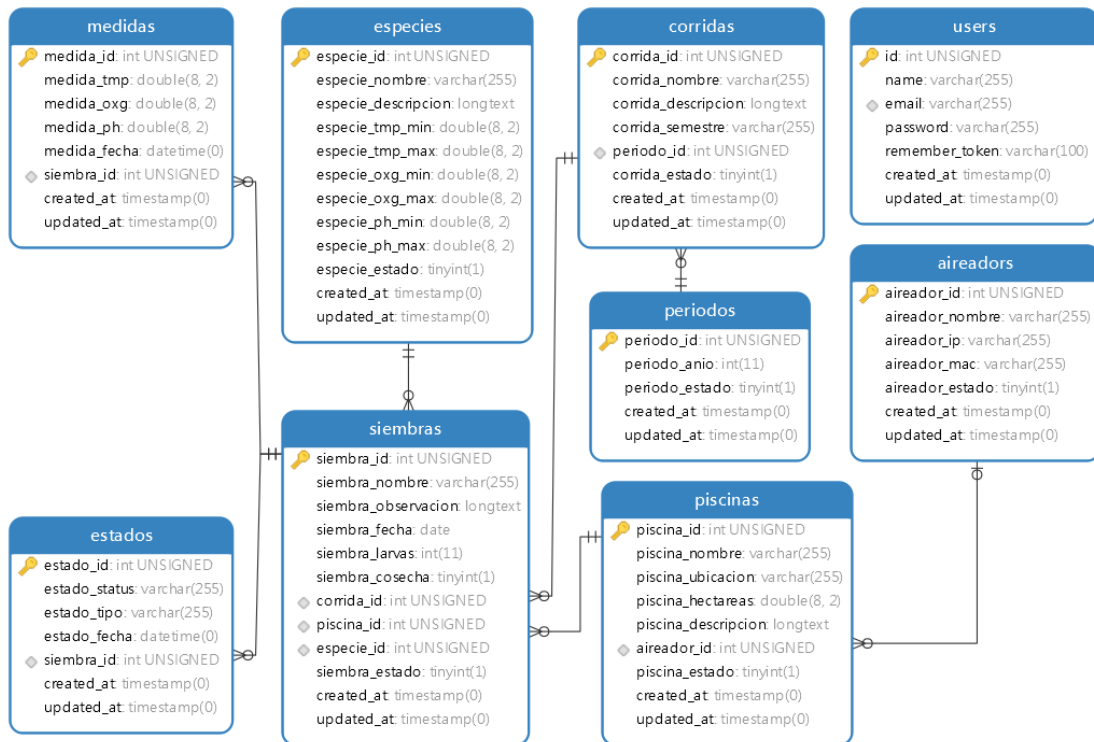
**Tabla 20** RF06 – Monitoreo

|                                    |  |
|------------------------------------|--|
| <b>Identificador del requisito</b> | RF06   |
| <b>Nombre del requisito</b>        | Monitoreo  |
| <b>Tipo</b>                        | Funcional  |
| <b>Descripción</b>                 | Permite al usuario visualizar las piscinas las cuales han sido o no sembradas en una determinada corrida seleccionada por el usuario. Es posible encender o apagar el sistema de aireación de las piscinas sembradas y a su vez monitorear en tiempo reales los valores medidos en cada uno de los parámetros de calidad del agua. |
| <b>Actores</b>                     | Usuario  |
| <b>Prioridad</b>                   | Alta   |
| <b>Condición de éxito</b>          | Se muestran las piscinas sembradas o no de una corrida.<br>Se apaga y enciende el sistema de aireación en las piscinas sembradas.<br>Se monitorea en tiempo real los valores medidos en cada uno de los parámetros de calidad del agua.  |
| <b>Condición de fracaso</b>        | No se muestran las piscinas sembradas o no de una corrida.<br>No se apaga ni enciende el sistema de aireación en las piscinas sembradas.<br>No se monitorea en tiempo real los valores medidos en cada uno de los parámetros de calidad del agua.  |

**Fuente:** Elaboración propia

## ANEXO E: Diseño relacional de la base de datos del software web SHIRMPSystem

Ilustración 3 Diseño de la base de datos del software web SHIRMPSystem



Fuente: Elaboración propia

## ANEXO F: Pantallas del software web SHIRMPSystem

Ilustración 4 Módulo de gestión de piscinas – Listar

SHIRMPSystem Un sistema para la gestión del proceso de aireación en la producción camaronera!!!

Listado de PISCINAS

Buscar...

| Id | Nombre           | Ubicación       | Hectareas | Aireador           | Opciones  |
|----|------------------|-----------------|-----------|--------------------|---|
| 4  | Piscina número 4 | Piscina 4 Sur   | 6         | Control Aireador 4 | <input type="button" value="Editar"/> <input type="button" value="Eliminar"/> |
| 3  | Piscina número 3 | Piscina 3 Sur   | 4         | Control Aireador 3 | <input type="button" value="Editar"/> <input type="button" value="Eliminar"/> |
| 2  | Piscina número 2 | Piscina 2 Norte | 3         | Control Aireador 2 | <input type="button" value="Editar"/> <input type="button" value="Eliminar"/> |
| 1  | Piscina número 1 | Piscina 1 Norte | 5         | Control Aireador 1 | <input type="button" value="Editar"/> <input type="button" value="Eliminar"/> |
| Id | Nombre           | Ubicación       | Hectareas | Aireador           | Opciones  |

Fuente: Elaboración propia

## Ilustración 5 Módulo de gestión de especies – Editar

SHIRMPSystem Un sistema para la gestión del proceso de aireación en la producción camaronesa!!!

### Editar ESPECIE

Nombre de la especie:  Descripción:

Temperatura:  Mínimo 10 Máximo 30

Oxígeno Disuelto (OD):  Mínimo 1 Máximo 10

pH:  Mínimo 6 Máximo 9

Fuente: Elaboración propia

## Ilustración 6 Módulo de gestión de corridas – Editar

SHIRMPSystem Un sistema para la gestión del proceso de aireación en la producción camaronesa!!!

### Editar CORRIDA

Nombre de la corrida:  Descripción:

Seleccione el periodo:  Seleccione el semestre:

Fuente: Elaboración propia

## Ilustración 7 Módulo de gestión de siembras – Registrar

SHIRMPSystem Un sistema para la gestión del proceso de aireación en la producción camaronesa!!!

### Registrar nueva SIEMBRA

Nombre/ Descripción de la Siembra:  Observación:

Seleccione la corrida:  Seleccione la piscina:

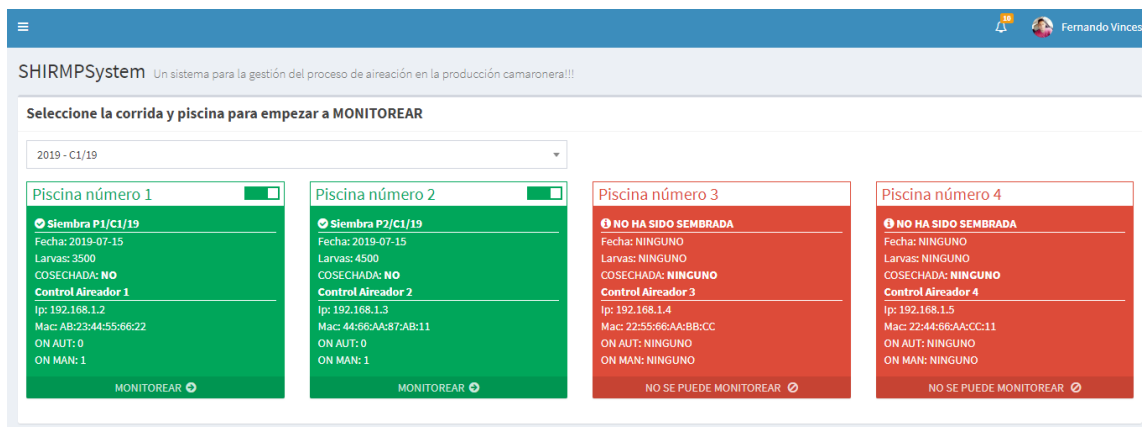
Seleccione el tipo de especie:  Número de larvas:

Fecha:  La siembra ha sido cosechada?:

Fuente: Elaboración propia

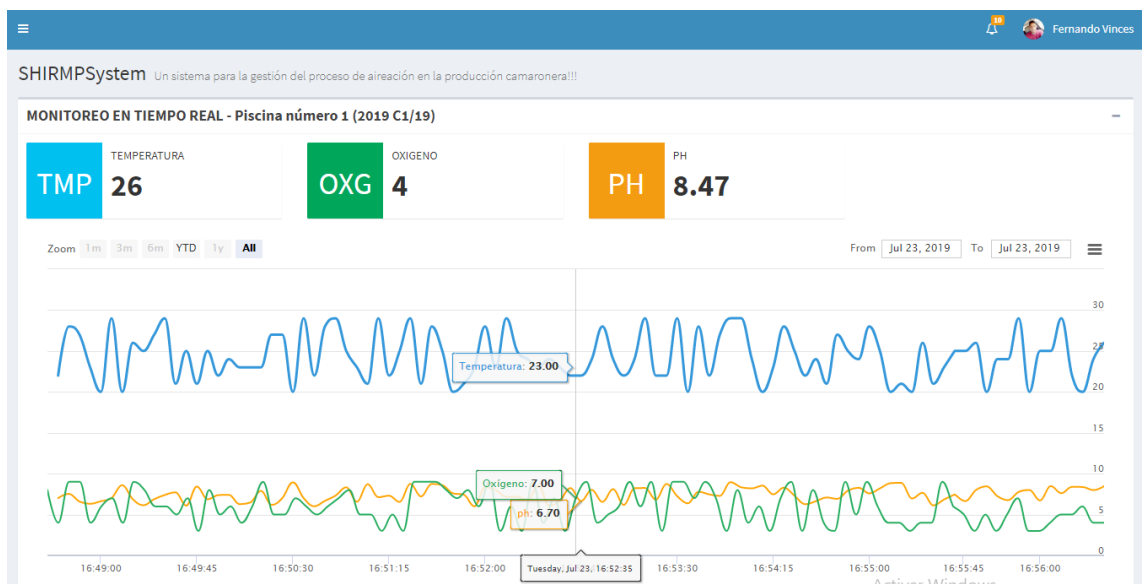


## Ilustración 8 Módulo de monitoreo – Visualización de piscinas



Fuente: Elaboración propia

## Ilustración 9 Módulo de monitoreo – Tiempo real



Fuente: Elaboración propia