

# Proposal of a Sustainable Agile Model for Software Development

Oscar Antonio Alvarez Galán<sup>1</sup>, José Luis Cendejas Valdéz<sup>2</sup>, Heberto Ferreira Medina<sup>3</sup>  
Gustavo A. Vanegas Contreras<sup>4</sup>, Jesús Leonardo Soto Sumuano<sup>5</sup>

Engineering in Information Technologies, Universidad Tecnológica de Morelia, Morelia Michoacán, México<sup>1,2,4</sup>  
Unit of Information Technologies and Communications, Instituto de Investigaciones en Ecosistemas y Sustentabilidad–UNAM  
TECNM/I.T. de Morelia, Morelia Michoacán, México<sup>3</sup>  
Centro Universitario de Ciencias Económico Administrativas (CUCEA). Universidad de Guadalajara (UdeG)<sup>5</sup>  
Cuerpo Académico de Tecnologías de Información y Comunicación (UDG-CA-125)<sup>5</sup>

**Abstract**—Sustainability is even more important now than ever if we speak in the context of organizational growth, it is necessary that technological products, such as software developments, are certified as green- environmental friendly technology that would mean a competitive advantage for an organization that implements an agile methodology for software development that takes sustainability into account, giving the organization new ways to market their software products as environmentally friendly. This study proposes a model for agile software development, it has taken into account that software development must be based upon reusing old hardware, free non-privative software and code (open source), as well as virtualization of servers and machines, to create software that can be useful for over a decade, as a result, we expect a reduction of planned obsolescence in hardware, which means taking one step ahead to help solve the problem that the big amount of electronic waste (e-waste) means nowadays worldwide.

**Keywords**—Sustainability; agile methodologies; software development; MDSIC; sustainability variables

## I. INTRODUCTION

Sustainability is more important every day if we talk about development processes at technological organizations. Today it is necessary that products and services are certified as green technology, there is data from Solving the e-waste problem (STEP) [1], that shows every person produces an estimated amount of 7 to 9 kilograms of electronic waste per year in Mexico, it is estimated that 42 billion tons of e-waste will be generated worldwide by 2016.

Technology companies generate their products and software and cloud services using traditional methodologies, these do not take into account economic sustainability or ecological sustainability, so devices that use this software have a predefined useful life for about 2 to 4 years approximately, it means they are fabricated with the idea that these devices, especially mobile ones will become obsolete within a short time, that way it increases the product demand [2], [3],[4], [5]. It has been shown that as time passes software becomes obsolete and can no longer update.

Companies that develop software require greater hardware resources so that their applications work in the best way, which generates a limited lifetime in the devices and the software that

is built requires more resources for its optimal operation; thus, advancing the useful life and becoming technological waste.

The development of custom made software represents high costs for organizations and many of these projects finally do not meet its minimum requirements. Organizations with different business lines want to enhance their information processes with the help of software, made by so-called software factories. Those factories can help systematize and improve the processes of organizations, with the use of ecotechnologies costs will be reduced and using code that is sustainable software can be developed faster.

It is for this reason that mobile software development brings new challenges to the software industry, because mobility, interconnection, and simpler applications are growing in demand. In many APPs global delivery stores, a tendency can be observed towards buying-selling software known as "freemium", a free license with minimum features, which can be changed by paying to access the software full features, upgrades and improvements.

It is for this reason that this research proposes a methodological model so agile software can be developed and be useful for at least 10 years, that way it can make a difference in the efficiency of the essential process of organizations. This research solves many issues related to how green software can be developed, the proposed methodology on this research takes into account measuring sustainability indicators and attributes and shows how those are related to each step of the agile development of mobile software. It also proposes a full architecture that uses microservices and non-relational databases, so that sustainable software development can be achieved. Most modern companies can certify this kind of development as green software which represents a competitive advantage for organizations.

The rest of the work is organized as follows – the next section discusses briefly sustainability, ecotechnologies and previous methodologies based on the experience of this industry in Mexico. This experience allowed building a methodological process, that takes into account attributes and indicators so we can be sure that technological products developed using the proposed methodology are sustainable. Also, the paper presents the results obtained from the

development of a software prototype and how we can measure the attributes and indicators on that software.

## II. LITERATURE REVIEW

The Earth Charter is a declaration of fundamental principles for the construction of a global society in the XXI Century [6], which is fair, sustainable and peaceful. It inspires people to understand the concept of shared responsibility. The charter means hope and it is a call for action, it contains information about how to transition to sustainable lifestyles and sustainable human development. As a consequence of human activity, climate change is manifesting itself inevitably, bringing consequences for future generations [7]; which are mainly observed in the increase in temperature and the increase in sea levels, as well as the loss of species worldwide.

One of the most important entities in the care of the environment worldwide, the international organization, World Wide Fund for Nature (WWF) [8] [9], Living Planet report, presents a series of graphs based on different studies such as world population, carbon dioxide (CO<sub>2</sub>), fertilizer consumption, freshwater use, tropical forest loss, marine fish capture, which show the effects caused on the planet, due to the production of electronic waste of the human being, which are observed in the Fig. 1.

With the data in the previous graphs, technological proposals must have the attribute of sustainability as a basis [10], this concept refers to the efficient and rational management of resources so that it is possible to improve the well-being of the current population without compromise the quality of life of future generations. On [11] it is mentioned that sustainability sciences study this concept and propose solutions from a perspective that integrates various disciplines and the treatment of problems that prevent the transit to sustainable development.

Ideally, technology and software products should have an acceptable degree of resilience, that is, the ability to endure over time and the changes to which they are subjected, this concept is important to apply when talking about sustainability and technology, however, it is common for companies to focus more on economic benefit, giving the product a predetermined period of life.

Companies or organizations that do not take care of sustainability and resilience in their technological products fall into a practice known as planned obsolescence which is a production policy, it manages the useful life of an object being determined so that the object is then obsolete, useless or no function in a short time [5]. This industrial policy is potentially beneficial for the producer since it forces the consumer to acquire the same product several times, stimulating demand for it.

It is for the reasons described above that organizations and companies need to apply ecotechnologies in the development of their technological products, thus promoting sustainability and resilience, preventing them from becoming obsolete in a short time.

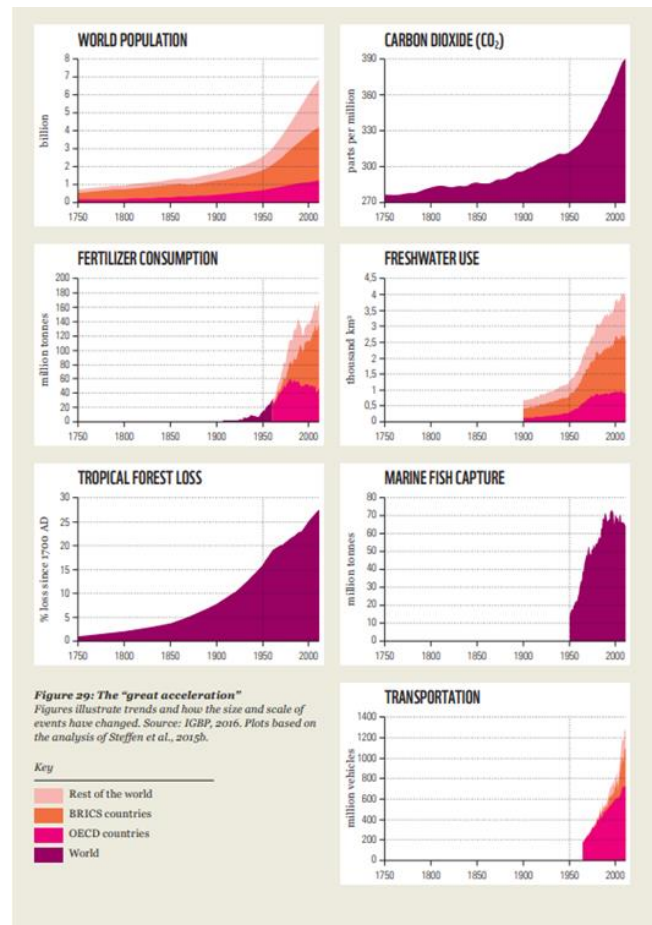


Fig. 1. Trends in Global Climate Change [8], [9].

It is mentioned that ecotechnologies “are the use of technological means for the management of ecosystems, based on a deep knowledge of the principles on which natural ecological systems are based and the transfer of this knowledge towards their management, so that the damage caused to the environment is minimized” [12] Among the most used green technologies today are solar panels, wind turbines, biogas chambers, energy-saving bulbs, solar heaters, dry toilets, etc.

It will be important that Information Technologies (IT) is seen in the organization as a great investment, since the use of technologies is a competitive differential for it, keeping IT up to date means a constant economic investment and Certifying technological products as sustainable can greatly help the return on investment. To achieve this denomination according to [13], [14] green technologies must meet certain criteria:

- Be accessible, especially to the poorest sectors of society.
- Be focused on local needs and contexts.
- Be friendly to the environment, promoting the efficient use of resources, recycling and reuse of products.
- Promote the use of local resources and their control.

- Generate employment in regional economies, especially in rural areas, from which the population has had to migrate due to a lack of opportunities.
- Preferably be produced on a small scale and in a decentralized manner.
- Be designed, adapted and disseminated through participatory processes, with dialogue between local knowledge and scientists.

It is for this reason that every implementation of technology in an organization should be aimed at cost reduction and that is why virtualization can be considered a green technology. It is mentioned on [15], [16] the need for departments or areas that are dedicated to generating the research, development and innovation process (R & D & I) is essential for most companies; this in order to enrich their processes. this department should ideally be in charge of applying a methodology that guarantees that the developed technological products have a useful life of at least 10 years and comply with the basic concepts of sustainability and resilience.

Among the existent models, on [15] it is proposed the Sustainable Technological Model (DESUSTEC), Which focuses on proposing a regulatory framework for the development of technology that is manufactured in software companies in Mexico. Being also sustainable, which is accompanied by good practices, is an alternative in the use of green technological infrastructure [15], [16].

As we can see in the Fig. 2 DESUSTEC consists of 6 levels, the first level consists of a software development cell made up of engineering students that will serve as a source of human resources for companies dedicated to software development, the second level refers to the use of Collaborative Integrated Model in Agile Software Development (MDSIC) which serves as a guide for the development of agile software, the third level refers to the use of ecotechnologies such as solar energy, virtualization and use of free software, the fourth level refers to the measurement of indicators and attributes of sustainability in the software, finally, the fifth level refers to the development of mobile software with a useful life of at least 10 years complying with the aforementioned indicators and attributes of sustainability, and the sixth level focuses on the reduction of electronic waste produced by mobile devices, servers, etc.

On the second level of DESUSTEC is contemplated MDSIC proposed in [17] which aims to develop quality software in small and medium software enterprises (SMEs), based on the alignment of information technologies with the essential process of the organization as well as being an agile model. MDSIC proposes five different levels that provide best practices for software development, in addition to being supported by the processes proposed by Project Management Institute (PMI).

MDSIC evaluates the quality of the software through a series of indicators that must be considered for optimal functioning. The first five elements that PMI mentions are integrated into the model: 1) Project integration, 2) scope, 3)

time, 4) cost, 5) quality. This will allow the measurement of quality in a project attached to the standards proposed by PMI.

Fig. 3 details each of the MDSIC levels, which will help the software developed to be aligned with the essential process of the organization and its levels are:

- Level 0: Problem Detection
- Level 1: Analysis and Design,
- Level 2: Development,
- Level 3: Implementation,
- Level 4: Quality indicators.

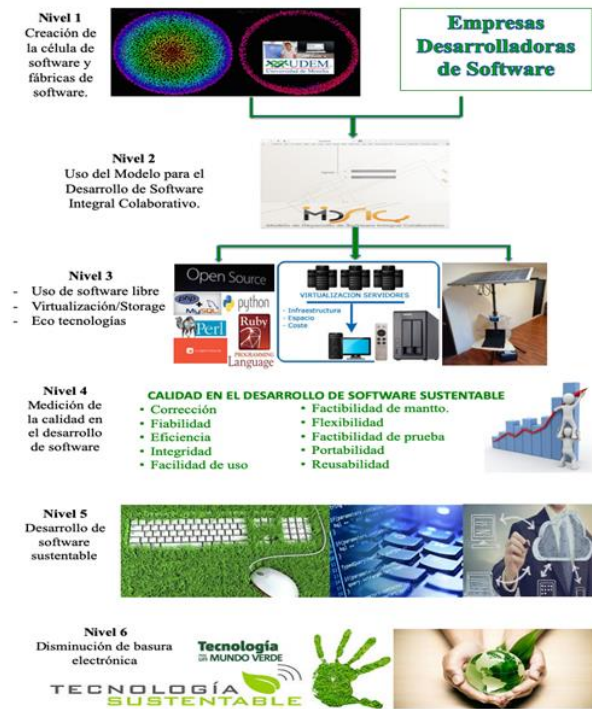


Fig. 2. DESUSTEC Model [15].

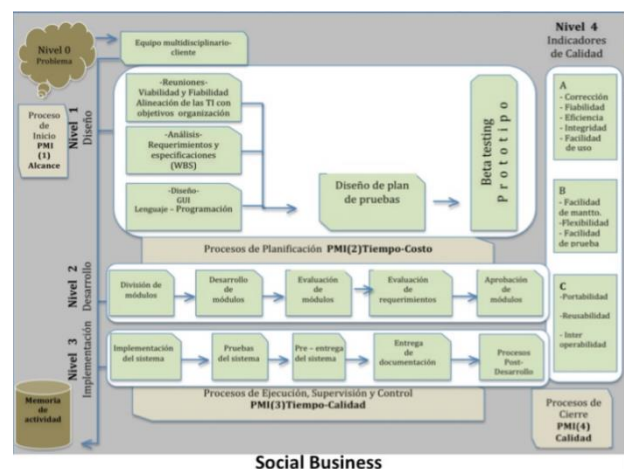


Fig. 3. MDSIC Model (Collaborative Integrated Model in Agile Software Development) [16].

Given the market trend towards the development of mobile technology [3] [18] proposes a version of MDSIC. Collaborative Integrated Model in Agile Software Development for Mobiles (MDSIC-M), it is a model based on MDSIC, for agile software development, which is oriented to rapid development and that meets the needs of small and medium companies, also proposes the documentation of the projects based on the model, through stories that facilitate the development of subsequent projects, as well as the use of their MDSIC 2.0 tool (software guide to apply the methodology in the development of a quality APP ).

MDSIC-M elements are:

- a) Establishment of requirements (tasks, deliverables and metrics),
- b) Initial plan or WBS (Work Breakdown Structure),
- c) Establish deliverables in the WBS,
- d) Use of a scheduler for task control (scheduler),
- e) Project time control by deliverable,
- f) Risk control when allocating resources,
- g) Cost control (establishment and metrics),
- h) Guarantee the quality; KPIs (performance indicators) and the Scope (total time) of the project,
- i) Use of a collaborative tool based on the scheduler,
- j) Test and delivery plan (prototypes) and
- k) Document lessons learned (database with project information).

The scheduler or planner is the core of MDSIC-M as it tracks and controls all the activities or tasks defined in the WBS or breakdown structure, specifies a list of activities with a hierarchy that allows building a complete map of all project activities, tasks, resources, times, costs and deliverables.

MDSIC-M proposes improvements to offer quality through KPIs; establishing roles, collaborative work teams, clearly defining deliverables, tools to use, standards to generate relevant information, control risks (costs) and finally ensure quality using techniques and standards. Fig. 4 shows MDSIC-M model.

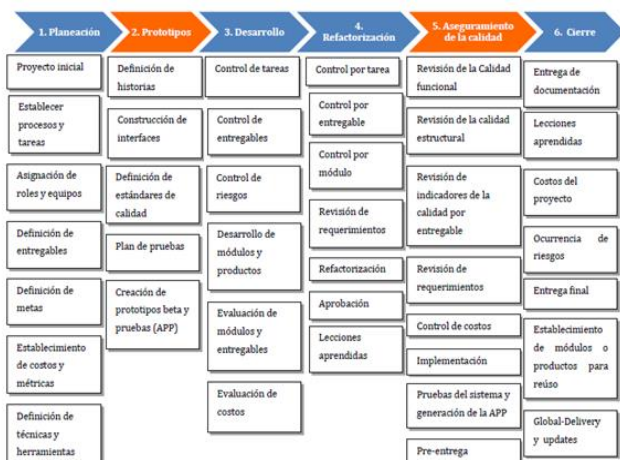


Fig. 4. MDSIC-M Model (Collaborative Integrated Model in Agile Software Development for Mobiles) [3].

Based on the aforementioned models and using the model proposed in this document, the degree of sustainability and resilience in the software will be measured through indicators and attributes, for this reason, the proposed model is also based on the Framework for the Evaluation of Natural Resource Management Systems Incorporating Sustainability Indicators (MESMIS) which incorporates sustainability indicators, MESMIS is a model that helps assess the sustainability of systems that manage natural resources, it defines a series of critical points or strengths and weaknesses for sustainability, which are related to three areas of sustainability: environmental, social and economic [19].

In this research, the proposal of a prototype of sustainable and agile mobile software was developed based on a methodological model that integrates the measurement of MESMIS sustainability indicators, together with the sustainability indicators obtained from the main correlations of this same study, with the objective that the proposed software prototype be sustainable and with a useful life of at least 10 years.

### III. METHODOLOGY

It was determined that the nature of the present investigation is of type a) analytical, as indicated [20] since facts and information that are already available can be used and this material was analyzed to perform a critical evaluation, b) exploratory, which is used when the research problem or topic is poorly studied and has not been addressed before. That is when the literature review reveals that there are ideas vaguely related to the problem of study [21], c) quantitative, is based on the measurement of quantities, it is applicable to phenomena that can be expressed in terms of quantity. d) non-experimental, it is done without deliberately manipulating variables, no situation is constructed, but situations are observed, it is not intentionally provoked, there is no control over the variables [20]. e) documentary, which defines [21] how to detect, obtain and consult bibliography and other materials that participate in other knowledge and / or information collected moderately in a selective way so that they can be useful to identify studies.

Table 1 shows how the research for the construction of sustainable software was carried out in 5 main stages which include 1) Characterization of sustainable software, where a background check was made regarding sustainability and development of software with green technology, 2) independent and dependent variables which will be measured, as well as to identify what were the sustainability variables not measured by MDSIC-M (Collaborative Integrated Model in Agile Software Development for Mobiles) [3], 3) Application of a survey to professionals of sustainability sciences and software developers, where the variables identified above were measured, 4) The proposal of a model, based on DESUSTEC in which the most important correlations obtained in the survey are analyzed and based on them defined the most important variables to be measured during the process of software development with MDSIC-M, 5) The creation of a software prototype for the measurement of these indicators, following the methodology, to guarantee the quality in the development.

TABLE. I. METHODOLOGICAL MODEL

<b>1.-Characterization of sustainable software development</b>
Characteristics of sustainable software development or identification of variables. Cross-sectional analysis of existing models and methodology.
<b>2.-Objectives of the methodology</b>
According to the objectives, establish the dependent and independent variables to analyze. Establish variables not measured using MDSIC.
<b>3.-Survey of professionals in sustainability sciences and sustainable development</b>
Analyze the acceptance and use of the elements of the DESUSTEC model in Mexico. Establish the population of the survey according to a confidence interval. Analyze the consistency of the surveys applied using Cronbach's alpha. Establish correlations between the census variables (Pearson).
<b>4.-Proposal of the methodology</b>
Analyze the correlations and establish the importance of each of the census variables. Establish the most important sustainability variables and their alignment with a development methodology. Propose a methodology, using standards and best practices.
<b>5.-Methodology tests</b>
Design the elements of the methodology and rely on the development of software to simplify its use. Perform a quasi-experiment to verify the feasibility of use. Document results.
<b>6.-Technology Transfer</b>
Perform a quasi-experiment to verify the feasibility of use. Document results.

<sup>a</sup>. Table 1.- Methodological model.

Teachers of the state of Michoacán were selected for this research, in [22] there are 8,848 teachers of higher education (population of professionals, whose lines of research focus on IT and the environment). From the data obtained previously and with a 95% confidence interval and a margin of error of 5% [20], according to the method of finite populations, the sample size was obtained, (1) was applied and the replacement of the data is shown in Table 2.

$$n = \frac{z^2 \cdot p \cdot q \cdot N}{(e^2(N-1)) + (z^2 \cdot p \cdot q)} \quad (1)$$

Where:

N = The size of the population to be registered (Higher education teachers in Michoacán).

z = Area under the normal curve with confidence interval: 95% q = 1 - p = 0.5 Proportion of the sample for a maximum n with greater precision.

p = % of times the phenomenon occurs in the population.

e = margin of error of the calculated sample (5%).

TABLE. II. REPLACEMENT OF DATA

Confidence level 95%	p= 95%	e = 5%
z= 1.96	(1-p)= 5%	N = 8848

n=72

Once it was carried out, it was analyzed, where the alpha Cronbach coefficient [23] was obtained, with a value of 0.737 which indicates an acceptable internal consistency in the survey results. The most significant Pearson correlations obtained are those with values >= 0.5, which are detailed in Table 3.

TABLE. III. MAIN CORRELATIONS

1.-Direct contact with users and constant feedback.
2.-Correct analysis of the problem to be solved and requirements.
3.-Free Software Implementation
4.-Use of eco technologies for saving electricity consumption.
5.-Reused materials and components (hardware)
6.-Efficient use of natural resources
7.-That the software on the devices is kept updated.
8.-Good management of risk management in planning.
9.-Guarantee the useful life of the software for at least 10 years of operation.
10.-Compatibility even on older machines or different operating systems.
11.-Support, connection with other applications, maintenance and speed, stand out as quality indicators for the software.
12.-Ease of use, utility, reliability, performance, and compatibility should be a priority to develop software.
13.-Proper planning and management of the same project must be carried out supported by tools to facilitate the tasks.
14.-Defining clear roles and objectives, each member of the development team must know what to do.
15.-Management of times previously established in the planning, to obtain software with quality, this must be developed in the times dictated from the beginning.
16.-Use of virtualization technologies that will allow us to use virtual versions of devices, meaning savings in energy consumption and hardware.

#### IV. DISCUSSION AND ANALYSIS

The proposed methodological model based on MDSIC-M is illustrated in Fig. 5.

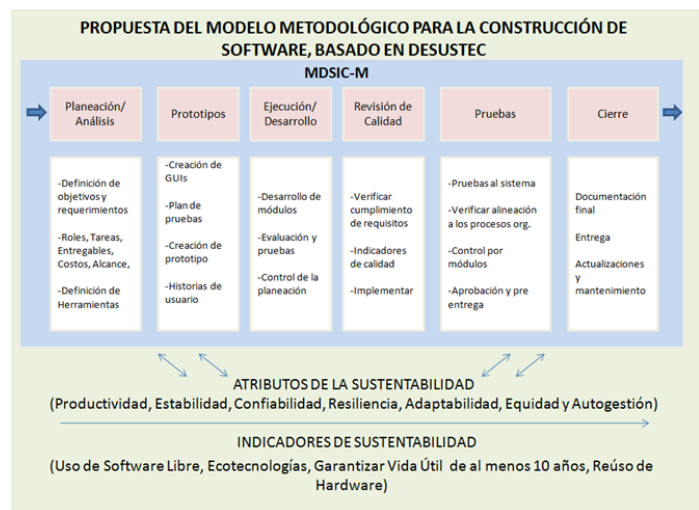


Fig. 5. Proposal of the Methodological Model for Software Construction, based on DESUSTEC.

Table 4 describes how each of the sustainability attributes based on MESMIS will be evaluated in the software and how it relates to the sustainability indicators obtained in the main correlations of this study.

TABLE. IV. ATTRIBUTES AND INDICATORS

Attributes	Equation	Related indicators	Equation
<b>Productivity</b>	The number of modules developed / Development time.	Life of at least 10 years of the software	10-year return on investment
<b>Stability</b>	The number of errors detected in tests / The Number of modules developed.	Use of free software	Total number of free software used / Total software
		Life of at least 10 years of the software	10-year return on investment
<b>Reliability</b>	Amount of money invested in development / Development time	Use of free software	Total number of free software used / Total software
<b>Resilience</b>	The number of functional modules / Number of years in operation	Use of free software	Total number of free software used / Total software
		Ecotechnologies	Number of applied ecotechnologies / Total applied technologies
		Life of at least 10 years of the software	10-year return on investment
		Hardware reuse	The number of reused hardware parts / Total hardware used
<b>Adaptability</b>	Frequency of use of ecotechnologies and reuse of software / number of years in operation	Ecotechnologies	Number of applied ecotechnologies / Total applied technologies
		Life of at least 10 years of the software	10-year return on investment
		Hardware reuse	Number of reused hardware parts / Total hardware used
<b>Equity</b>	The number of users / The number of free software used	Use of free software	Total number of free software used / Total software
		Ecotechnologies	Number of applied ecotechnologies / Total applied technologies
		Hardware reuse	Number of reused hardware parts / Total hardware used
<b>Self-management</b>	Total current cost of software development / Initial cost of software development	Life of at least 10 years of the software	10-year return on investment

It is proposed as good practices for software construction to follow sustainability indicators, which allow to generate a uniform architecture using [24], [25], [26] for the layers of the applications, separating them into microservices that are implemented in an API as a communication protocol between the client and the server, facilitating its maintenance, scalability, and compatibility in the software through these sustainable indicators. Table 5 shows the proposed layers.

The prototypes developed in future works based on this model will measure attributes and indicators using the equations already shown, and it will be known the level of sustainability that each software development has, so companies keep using it every time they develop new software [27], [28].

An example of the application of the model proposed was built as a prototype that shows a CRUD (Create, Read, Update, Delete) for a general non-relational database, where the use of the code is shown through generic libraries in Swift. For this application, the indicators and attributes used in the proposed model were used.

## V. RESULTS

The software prototype built using this model as a base uses functions that allow receiving any number of parameters in an array which is part of the data sent by the client in a request to the server to insert, update, delete or request data, is generated in this way so that the code is reusable for any context, the prototype allows the names of the keys and values to be assigned to convenience, open-source programming language swift 3 was used for this prototype as well as [29], [30], [31] the uniform architecture for sustainable software development proposed on Table 5.

TABLE. V. UNIFORM ARCHITECTURE FOR SUSTAINABLE SOFTWARE DEVELOPMENT

API (Application) <ul style="list-style-type: none"> <li>Interaction with client / server by HTTP protocol allows interaction between both</li> </ul>	Interface layer at the protocol level. In the prototype developed, the API interacts with the server through HTTP routes, through which REST requests can be sent
Frontend (View) <ul style="list-style-type: none"> <li>It is the interface between the user and the system, implements interaction techniques and methods. Construction tools with good practices are taken into account.</li> </ul>	Client side layer. It is the graphic interface of the prototype developed
Middleware (intermediary) <ul style="list-style-type: none"> <li>It is an intermediate software layer between the operating system and communication networks (services), below the user's applications</li> </ul>	The intermediate layer of communication between client and server
Backend (base software) <ul style="list-style-type: none"> <li>It is the information access layer that defines the server-side database</li> </ul>	Server side layer. A server that handles the data in the prototype, for this development the NOSQL server Apache Couch DB (Apache, 2016) was used.

Based on [25], [26].

Fig. 6 shows some examples of the functions used within the software prototype:

```
func insert() {
    items.append("Item \#{items.count + 1}")

    let insertionIndexPath = IndexPath(row: items.count - 1, section: 0)

    tableView.insertRows(at: [insertionIndexPath], with: .automatic)
}

override func tableView(_ tableView: UITableView, numberOfRowsInSection section: Int) -> Int {
    print("NUMERO DE ITEMS-\#{items.count}")
    return items.count
}

override func tableView(_ tableView: UITableView, cellForRowAt indexPath: IndexPath) -> UITableViewCell {
    let myCell = tableView.dequeueReusableCell(withIdentifier: "cellId", for: indexPath) as! MyCell
    myCell.nameTextField.text = items[indexPath.row]
    print("VALORES DE ITEM-\#{items}")
    myCell.myTableViewController = self

    return myCell
}

override func tableView(_ tableView: UITableView, viewForHeaderInSection section: Int) -> UIView? {
    return tableView.dequeueReusableCell(withIdentifier: "headerId")
}
```

Fig. 6. Software Prototype Functions, based on the Proposed Methodological Model using Swift Code [32].

In Figure 7 and 8 two interfaces of the software prototype developed during the present investigation are observed.



Fig. 7. Software prototype login screen, based on the proposed methodological model.

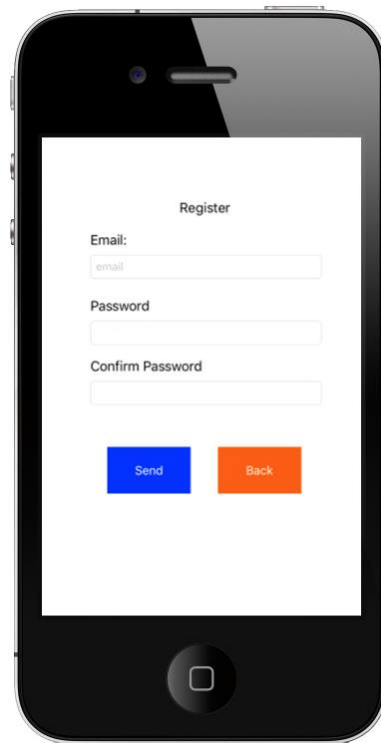


Fig. 8. Software Prototype, Register Screen.

## VI. CONCLUSION

According to the data and calculations of sustainability indicators obtained, it can be seen that there is still work to be done so that the model improves, along with how sustainability is ensured in software development. Since the measurement of indicators in the proposal depends on the number of years elapsed and how productive the software remains after these; understanding that sustainability in software is just being explored since it is a new issue and there is still much more to understand and improve in this field.

According to the revised agile methodologies for the model, none takes sustainability into account, so this represents an advantage over others in its application. Software developed under the proposed model may be marketed as green technology. Saving energy is encouraged, implementing ecotechnologies such as virtualization for the use of microservices which represents a competitive advantage for organizations, as well as the reduction in production costs and time. Since one of the main pillars of the proposal is the reuse in code to speed up the creation of the software. It is intended to generate in a future project a uniform infrastructure that allows sustainably to offer the creation of microservices [28], offering SaaS (software as a service), [29], [30], a uniform architecture that encompasses all the technologies that today are scattered, helping users to develop software quickly with quality. The proposed model may be a sustainability framework applicable to technological projects, even in other areas of knowledge. Today the legal framework, begins to request in several countries that companies contemplate ethically and socially to the sustainability in its processes for the generation of products and / or services.

#### ACKNOWLEDGMENT

We are grateful for the support of the Ecosystem Research Institute, to the Technological University of Morelia. Especially to Eng. Atzimba Graciela López Maldonado., MTI. Alberto Valencia García And Hugo A. Zavala V. his comments and support in the analyzes. To Eng. Diego Cabrer F. for his help in the application of the model.

#### REFERENCES

- [1] STEP, «Solving the E-wast problem» 2016. ISSN:2071-3576.
- [2] Asociación Mexicana de Internet, «Dispositivos Móviles: Hábitos del consumidor Mexicano,» 2014. [Online].
- [3] H. Ferreira, «Metodología MDSIC-M para el desarrollo de aplicaciones móviles en empresas Pymes,» 2015. [Online].
- [4] K. Brigden, I. Labunska, D. Santillo, & M. Allsopp, Greenpeace, «Recycling of electronic wastes in China and India: Workplace and environmental contamination,» 2005. [Online].
- [5] P. Ramírez, «Obsolescencia tecnológica programada,» Facultad de ciencias y tecnología, Universidad Católica Sede Regional Asunción, 2012.
- [6] Iniciativa Carta de la Tierra, «Carta de la Tierra,» 2016. [Online].
- [7] G. Gaidajis, K. Angelakoglou y D. Aktsoylou, «E-waste: Environmental Problems and Current Management,» 2010. [Online].
- [8] World Wide Fund for Nature, «Planeta Vivo Informe 2016» 2016. [Online]. ISBN: 978-2-940529-48-3.
- [9] Elsevier, «Sustainability Science in a Global Landscape,» Elsevier's Analytical Services, 2015.
- [10] G. Merino, R. Tovar « Gasto, inversión y financiamiento para el desarrollo sostenible en México,» Proyecto CEPAL/PNUD RLA/01/001 "Financiamiento para el Desarrollo Ambientalmente Sostenible en América Latina y el Caribe", 2002. ISBN: 9213220995.
- [11] T. Ortiz, A. Quiroz, C. Migoya, « Evaluación de Sustentabilidad Para el Manejo de Sistemas Naturales Manual Para el Técnico Comunitario » Basado en el Marco de Evaluación de Sustentabilidad: MESMIS 2016. [Online].
- [12] L. Gonzaga Gutiérrez, «Perspectivas de la biotecnología en las ecotecnologías,» Scientia et Technica, p. 451, 2006.
- [13] J. A. Ortiz, O. Masera y A. F. Fuentes, La Ecotecnología en México, México D.F.: Imagia, 2014.
- [14] C. Ferrete Sarria, «La eco-tecnología y la ciencia ecológica como alternativas. una larga historia de olvidos,» 1999. ISSN 1132-8177.
- [15] J. L. Cendejas Valdez, H. Ferreira Medina, G. A. Vanegas Contreras y M. A. Acuña Lopez, « El modelo sustentable tecnológico (DESUSTEC) como herramienta en el desarrollo de software para la reducción de basura electrónica » Journal of Technology and Innovation, p. 5-15, 2019.
- [16] T. Mackey, J. K. Benedict « XenServer Administration Handbook: Practical Recipes for Successful Deployments,» 2016. ISBN 13: 9781491935439.
- [17] J. L. Cendejas Valdéz, C. A. Vega Lebrún, A. Careta Isordia, O. Gutiérrez Sánchez y H. Ferreira Medina, «Diseño del modelo integral colaborativo para el desarrollo ágil de software en las empresas de la zona centro-occidente en México,» Revista Electrónica Nova Scientia, p. 133 – 148, 2014.
- [18] H. Ferreira, J. L. Cendejas Valdez, H. Ferreira Medina, G. A. Vanegas Contreras y G. Cortes Morales, A. H. Ginori González «Collaborative Integrated Model in Agile Software Development (MDSIC/MDSIC-M)-Case Study and Practical Advice » 2019. [Online].
- [19] M. Astier, O. Masera, L. García, C. González, S. López-Ridaura, T. Ortiz y Y. Galván-Miyoshi, «MESMIS - Marco para la Evaluación de Sistemas de Manejo de recursos naturales incorporando Indicadores de Sustentabilidad,» 2012.
- [20] C. Kothari, Research Methodology : Methods And Techniques, New Delhi: New Age International, 2019. ISBN-10: 9386649225.
- [21] R. Hernández, C. Fernández y L. Baptista, Metodología de la investigación, México: McGraw Hill, 2000.
- [22] Secretaría de Educación Pública Dirección General de Planeación, Programación y Estadística Educativa « Estadística educativa Michoacán de Ocampo Ciclo escolar 2018-2019, » 2018.
- [23] L. J. Cronbach, «Coefficient alpha and the internal structure of tests,» , pp. 297–334, 1951.
- [24] Apache, «Apache HTTP Server Implementation Notes,» 2016. [Online]. Available: <https://iiif.io/api/annex/notes/apache/#table-of-contents>. [Accessed: 06 Apr 2017].
- [25] K. Ayala, H. Ferreira, R. Ferreira, A. Nuñez y J. Ruíz, «Caracterización de WebApp para el desarrollo web - Arquitectura uniforme,» 2017.
- [26] W3C, «Web Services Architecture,» 2004. [Online]. Available: <https://www.w3.org/TR/ws-arch/wsa.pdf>. [Accessed: 10 Sep 2016].
- [27] P. R. Fillotrani, «Calidad en el Desarrollo de Software - Modelos de calidad de software,» 2007. [Online].
- [28] G. Sunkel, D. Trucco, A. Espejo, La integración de las tecnologías digitales en las escuelas de América Latina y el Caribe. Una mirada multidimensional, Santiago de Chile: Naciones Unidas Cepal, 2014. ISBN 978-92-1-121851-0.
- [29] M. Fowler, «Microservices: a definition of this new architectural term,» 2014.
- [30] Microsoft, D. Sprott y L. Wilkes, «Understanding Service-Oriented Architecture,» 2004.
- [31] M. D. Gonzalez Rio, Tecnologías de Virtualización, Createspace, 2014.
- [32] Apple Inc. (2014). The Swift Programming Language. Apple Inc.