

***Seminario de Lenguajes***

*“PLACEHOLDER* -> Metodología de trabajo de empresas aplicadas a un juego de computadora <- *PLACEHOLDER*”

**Docentes:**  
Hernán Merlino

Federico Ribeiro

**Alumnos:**

-Bombara, Martin Nicolás.

-Contento, Guido Ezequiel.

-D'Alessandro, Stefano Tomas.

-Montaña, Ignacio.

**RESUMEN**

Las metodologías de trabajo de las empresas han evolucionado hasta lo que hoy en día se cree es la forma más eficiente de trabajo posible. Entre los elementos adoptados por éstas encontramos las metodologías ágiles, y dentro de éstas, los distintos marcos de trabajo (Scrum, Kanban, etc), que utilizada en conjunto con sistemas de control de versiones, permiten facilitar y promover la efectividad de sus miembros, asi como lograr un proyecto de mayor calidad. A la par de esto avanzan rápidamente las técnicas de programación, lo que hace que haya que adaptarse a estos cambios, aprendiendo un lenguaje de programación de alto nivel y a trabajar con sus sentencias y librerías (Python en este caso).

Este trabajo tiene como objetivo aplicar estas metodologías pero en el marco de un juego de computadora con la finalidad de comprender si éstas verdaderamente son efectivas como metodología de trabajo, además de que podría mejorarse de las metodologías, para asi optimizar el desarrollo de un proyecto.

**ABSTRACT**

Working methodologies on enterprises have evolved until what nowadays is believed to be the most efficient way of working possible. Between these elements adopted by them, we find agile methodologies, and, as part of them, different frameworks (Kanban, Scrum, etc), which used together with version control systems, facilitate and promote the effectiveness of their members. Together with this, programming techniques advance quickly, forcing people to adapt to these changes, learning a high level programming language, and to work with its sentences and libraries (Python in this case).

The objective of this paper is to apply the aforementioned methodologies but in the development of a computer game, with the purpose of comprehending if they are effective as a working methodology, as well as understanding what could be improved to them, to optimize the development of a software project.

**ÍNDICE**

**1. INTRODUCCION 1**

**2. ESTADO DE LA CUESTION 2**

**3. PROBLEMA PRESENTADO 3**

**4. SOLUCION 4**

**5. CONCLUSION 5**

**6. FUTURA LINEA DE INVESTIGACION 6**

**7. AGRADECIMIENTOS 7**

**8. REFERENCIAS 8**

**INTRODUCCION**

Para comenzar, nos pondremos en contexto con lo que son las metodologías agiles usadas para el desarrollo de software en la actualidad, además de los elementos que se usan en conjunto con ellas (Controladores de versión). Luego hablaremos acerca de Python, el lenguaje elegido para llevar a cabo este proyecto, y de la librería utilizada para el desarrollo del juego.

Las metodologías agiles para el desarrollo de software aparecieron debido a los requerimientos cambiantes de la actualidad (es decir las constantes modificaciones que se deben realizar a un proyecto para asegurarse de que cumpla las expectativas), además de la falta de experiencia de los trabajadores de las empresas. Estas son **Kanban, Scrum y Continous Build**.

Kanban es un organizador de tareas, un sistema virtual que limita el trabajo disponible, encargado de mostrar las tareas a realizar divididas en columnas, para que cada miembro del equipo pueda elegir que tarea llevar a cabo, lo que produce cambios graduales y evolutivos al sistema, además de brindarle poder a quienes trabajan en el proyecto.

Scrum es una metodología que apunta a terminar un proyecto lo más rápido posible. Consta de pequeñas iteraciones de no más de 2 semanas, denominadas “sprint”, para luego, en una reunión, discutir acerca del desarrollo del proyecto y los problemas que se hayan tenido (se busca resolverlo en menos de dos horas).

Continous Build es una consecuencia del uso de estas metodologías. Consiste en compilar el código luego de cada sesión de trabajo, y dejarlo realizando pruebas. Los programas que permiten esto son Ant, un automatizador de pruebas, y Scheduler, como su nombre lo indica, un organizador.

Para lograr una buena gestión del proyecto, se utilizan los controladores de versiones junto con estas metodologías. Estos son sistemas que nos permiten tener varias versiones un mismo proyecto, para que cada miembro trabaje en una rama haciendo las modificaciones que desee.

Python es un lenguaje creado a finales de los años ochenta, de alto nivel, y es multiparadigma, es decir que aplica tanto al paradigma de programación funcional, como al paradigma orientado a objetos. Además, fue creado como un lenguaje que tenga una sintaxis que permita que sea legible fácilmente. Por estas razones, junto con el hecho de que sea código abierto, han aportado a que sea uno de los lenguajes más utilizados en la actualidad, para diversas implementaciones, entre ellas juegos de computadora, siendo algunos ejemplos concretos de esto Metin2 o Frets on Fire.

La librería utilizada para llevar a cabo estos proyectos es Pygame, que posee diversos módulos que facilitan la creación de juegos mediante el manejo de sprites (pequeños dibujos en mapas de bits para representar personajes, elementos, etc.).

**ESTADO DE LA CUESTION**

Como primera instancia, nos pondremos en contexto acerca de las metodologías de desarrollo de software previamente explicadas, hablando de la metodología más utilizada previamente a la aparición de las metodologías ágiles, explicando sus características, para luego detallar un poco más de por que se reemplazó por las metodologías ágiles. Luego, hablaremos del sistema de control de proyecto trabajado con las metodologías, para terminar explicando brevemente el trabajo que debimos realizar.

Previa a la utilización de las metodologías agiles, las empresas optaban por utilizar la metodología de Cascada (o Waterfall), una metodología muy estructurada en su aplicación, lo que llevaba al inconveniente de que si surgía la necesidad de algún cambio importante o que abarque muchas áreas del proyecto, obligaba a comenzar el proyecto desde cero, lo que resultaba en un gran gasto económico y de tiempo.

En este marco se da la aparición de las metodologías ágiles. Debido a su flexibilidad ante cambios o dificultades presentadas en el desarrollo, permitía adaptar el proyecto a estos cambios. Otra característica muy importante de sus marcos de trabajo, es la autogestión de desarrolladores del proyecto: entre los miembros se definen los tiempos y se hace la distribución de tareas del mismo, logrando que quienes aporten al proyecto sean los que tengan el mayor control del mismo.

Estas metodologías usualmente se trabajan en conjunto con los llamados controladores de versiones: sistemas en línea que permiten tener varias versiones del mismo proyecto software (denominado “repositorio”), donde los miembros del proyecto trabajan en “ramas”, (“branches” en inglés) realizando sus distintos aportes. Uno de los más utilizados es GIThub.

En nuestro caso, se nos presentó el desafío de construir un juego de computadora utilizando las metodologías y sistemas previamente mencionados y explicados, además de utilizar un lenguaje con el que no estábamos familiarizados, pero que resultó bastante simple de comprender debido a sus características principales.

**PROBLEMA PRESENTADO**

Para comenzar el trabajo, debíamos familiarizarnos con un lenguaje con el que no habíamos trabajado antes, pero que, como se explicó anteriormente, posee una sintaxis que lo convierte en un código fácilmente legible y comprensible. Python resultó ser un lenguaje bastante fácil de entender y de aplicar, ya que los problemas principales que se nos presentaron a la hora de programar solían ser problemas de identación (es decir, problemas a la hora de separar correctamente el código), ya que Python es muy estricto en este sentido, a diferencia de otros lenguajes.

Junto con esto, debíamos utilizar las metodologías ágiles ya mencionadas. En las clases teóricas de Seminario de Lenguajes fuimos capacitados acerca de ellas, es decir, de cómo están eran utilizadas en la mayoría de empresas en la actualidad, además de aprender cómo es la forma de trabajo de éstas, el cual se divide en varios ambientes de desarrollo donde se realiza preproducción, producción y testeo de un producto.

El trabajo a realizar consistía en poner en práctica lo aprendido en las clases teóricas, para que nosotros desde nuestro punto de vista de alumnos pudiéramos simular las futuras experiencias que tendríamos trabajando en una empresa.

Luego de investigar acerca de las herramientas con las que teníamos que trabajar, optamos por hacer un RPG, es decir un juego donde el jugador toma control de un personaje que posee ciertas características, y lo utiliza durante el transcurso del juego. Definimos que este tipo de proyecto sería lo más fácil a realizar dado el tiempo que teníamos y las herramientas a utilizar. Al mismo tiempo, establecimos todos los elementos que íbamos a necesitar para la construcción del juego.

Palabras Clave que rotulan el trabajo:  
**Python,   
Pygame,   
Libreria,  
Metodologias Agiles,   
Kanban,   
Scrum, Backlot, GIThub,   
Controlador de Versiones, RPG.**

**SOLUCION**

Como primera instancia para comenzar a trabajar, teníamos que aprender a programar en lenguaje Python. Para ello, decidimos recurrir a internet porque coincidimos que era la mejor fuente donde podíamos aprender. Dedicamos una semana a aprender el lenguaje, para luego hacer lo mismo con la librería Pygame.

***Aca contar la historia de cómo fue hacerlo (chamuyar a full)***

Como se dijo anteriormente, la utilización de las metodologías agiles facilto mucho encarar el proyecto.

Durante estos 3 meses de desarrollo, la creación del videojuego paso por distintas etapas.

La primera etapa fue la de “capacitación” en Python y Pygame, tuvimos reuniones en las cuales probamos distintos programas o códigos que escribimos y planteamos las dudas que surgían a raíz de esto. Ademas (*fin de la inspiracion)*

**🡪ESTADO DE C ESCRITO POR TINCHO// ME GUSTA MAS COMO PRESENTACION DEL PROBLEMA.** 🡨 Lo primero que debíamos hacer era aprender más acerca de Python y Pygame, lenguaje el cual hasta el momento no había utilizado ninguno de los integrantes pero al estar familiarizados con la programación en sí, no tuvimos mayores dificultades al aprenderlo ya que es un lenguaje muy intuitivo y de alto nivel.

Al momento de comenzar con el desarrollo lo primero era realizar la distribución de las tareas, para esto también tuvimos que familiarizarnos con Kanban, el cual tampoco tuvo mucha complejidad en ser utilizado.

Decidimos que todos deberíamos aportar algo en cada tramo del proyecto, tanto de documentación como de programación. Siempre intentando desarrollar bajo el marco de trabajo solicitado.

Consideramos que para el desarrollo de videojuegos, lo más eficiente es utilizar metodologías agiles ya que siempre se presentan cambios o se quiere agregar algo al programa y de este modo no se complica agregar estas modificaciones mientras se está programando el juego. Además, el hecho de poder autogestionar las tareas fue muy útil ya que cada uno tomo la tarea con la cual se sentía mas cómodo o era su “punto fuerte”