

***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 qué 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,   
Librería,  
Metodologías 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. La forma de aprendizaje fue mediante tutoriales de YouTube, ya que descubrimos que era la forma más didáctica y simple, y que cualquier duda que teníamos podíamos recurrir a esos videos de nuevo para aclararlas.

Una vez que entramos en terreno del lenguaje, procedimos a organizar el trabajo. Comenzamos por entender como tenía que ser el desarrollo del juego, ya que era la primera vez que hacíamos un trabajo de esta índole. Realizamos una investigación en internet acerca de cómo organizar el trabajo para este tipo de videojuego, y comenzamos asignando las responsabilidades de los miembros del equipo.

Al principio, comenzamos por hacer la base que necesitaríamos para hacer el código. Comenzamos con los mapas que tendrían el juego, y los sprites que usaríamos como “placeholders” para representar nuestros personajes antes de tener los sprites finales definidos. Con estos, comenzamos a escribir el código que podíamos en Python, para tener un poco de trabajo adelantando. Esto incluye el menú principal, la pausa que tendría nuestro juego y el Loop principal requerido por Pygame para leer los eventos que se lleven a cabo (tal y como habíamos aprendido en los tutoriales por los que optamos.)

Luego de esto, comenzamos a definir los sprites del juego, ya que los necesitábamos para escribir la siguiente parte del código (los movimientos de estos, las clases respectivas de cada uno, las colisiones con los elementos del mapa, los ataques que estos realizarían). En esta etapa fue en la que surgió más problemática, ya que requerimos de varias revisiones y prueba y error para lograr que el código funcione óptimamente. A la par de esto, algunos miembros del grupo se encargaban de la parte sonora del juego, buscando música para los diferentes mapas que habíamos creado y los efectos de sonido para los ataques que realice nuestro personaje.

Finalmente, comenzamos a trabajar en funciones específicas de nuestro código, esto incluye las interacciones con los enemigos del personaje o las transiciones entre los mapas que habíamos creado, para luego realizar un testeo y asegurarnos de que todo funcionaba correctamente.

**CONCLUSION**

La realización del proyecto no resultó tan complicada como nosotros esperábamos. Como ya se mencionó, esta era la primera vez que estábamos frente a la situación de construir un software por nuestra cuenta, y desde cero. Ciertamente durante la construcción del mismo tuvimos problemas, pero todos estos logramos solucionarnos.

Cabe destacar que aprendiendo las metodologías agiles y aplicándolas al trabajo nos ayudó bastante en lo que respecta a organización, ya que todo el tiempo sabíamos que era lo que el trabajo que teníamos que realizar (dado que en la tabla Kanban teníamos siempre una lista de tareas) y eso nos permitió en cualquier momento dedicarnos de lleno a una tarea, en caso de que demos por completa la tarea que teníamos asignada, o de ayudar a algún miembro del equipo si tenía dificultades con la que estaba realizando en ese momento.

La utilización del repositorio y el controlador de versiones también facilitó mucho el trabajo, ya que nos permitió que en todo momento los miembros del grupo tengamos una versión actualizada de lo que se venía haciendo, lo que nos brindaba integridad y seguridad a la hora de agregar los cambios que generaba cada uno. Además, dadas las características de estos sistemas, podíamos agregar cambios en todo momento y en cualquier lugar, y lográbamos que el resto del grupo se mantenga al tanto de los cambios generados, pues el sistema guarda un log con ellos.

Podemos concluir que la utilización de metodologías agiles es una gran elección a la hora de realizar un proyecto de software de ésta índole, ya que permite tener un control constante del proyecto que se está realizando, y acelera bastante la finalización del proyecto, pues logra maximizar la efectividad de los involucrados en el trabajo.

**FUTURA LINEA DE INVESTIGACION**

Logramos terminar el proyecto, pero aún asi existen diferentes elementos que no logramos adicionar debido a la falta de tiempo. En primer lugar, planeamos un sistema de objetos para que el personaje pueda utilizar, los cuales no pudimos implementar por falta de tiempo y dificultades al encontrar los atributos gráficos para reasentarlos en el juego.

Otro elemento que quisimos agregar fue un sistema de tiendas donde el jugador recolecte oro soltado por los diferentes enemigos, y con ellos lo que haga es adquirir mejoras para sus atributos en la ciudad principal, para facilitar el avance del juego y hacerlo más rápido.

Finalmente, habíamos planeado tener dos enemigos principales, donde el jugador debía primero derrotar a uno en un mapa de hielo, para conseguir el nivel suficiente para derrotar al enemigo final principal del juego, pero por una cuestión de tiempo solo llegamos a implementar el ultimo jefe final mencionado.

**REFERENCIAS**

Página de descarga de Python: <https://www.python.org/>

Página de descarga de librería Pygame: <https://www.pygame.org/>

Mapas realizados con Tiled Map Editor: <http://www.mapeditor.org/>

Música y efectos de sonido obtenidos de:

VGMusic: <https://www.vgmusic.com/>

YouTube: <https://www.youtube.com/>

Sprites obtenidos de: <http://spritedatabase.net/>

**AGRADECIMIENTOS**

Queremos agradecer principalmente a ChelinTutorials (<https://www.youtube.com/channel/UCZJS-lpC1BhLSdsjAqj1i8A>) por enseñarnos el lenguaje Python y a utilizar la librería Pygame, a las páginas mencionadas en las referencias por los elementos que utilizamos en el juego (ya que son de libre uso) y a la página Stack Overflow (<https://stackoverflow.com/>) por resolver las dudas que nos surgían a la hora de codificar.