|  |  |
| --- | --- |
| codEstudiante | Sebastián Achondo Silva |

|  |  |
| --- | --- |
| Supervisor (a) | Alonso Casas |

Fecha (Semana 1): 02/01/2024 – 05/01-2024

|  |
| --- |
| En la primera semana, se le otorgó una cálida bienvenida al alumno a la empresa, y se le introdujo al entorno corporativo, siendo conscientes de su limitada experiencia previa en el sector financiero. Para mitigar esta brecha de conocimiento, se organizó una serie de sesiones de inducción orientadas a familiarizar al alumno con los conceptos y terminología propia de la industria financiera. Estas actividades formativas se complementaron con la asistencia a diversas reuniones estratégicas, permitiendo así observar la aplicación práctica de dichos conceptos dentro de un entorno real de negocio.  Paralelamente al conocimiento teórico, se efectuó una inmersión en herramientas y *software* esenciales, procediendo con la descarga e instalación de los programas requeridos para el desempeño de sus funciones en la empresa. A nivel técnico, como cometido inicial, se asignó la tarea de desarrollar un código en *C#*, cuyo propósito era la extracción de transacciones del Banco Security.  El software desarrollado este lenguaje de programación dadas sus sólidas capacidades para interactuar con bases de datos y realizar operaciones de entrada/salida de datos complejas. El código resultante fue diseñado para conectarse de manera eficiente a la entidad financiera, extraer la información pertinente de transacciones bancarias y almacenarla adecuadamente en una base de datos alojada en *SQL Server Management Studio*. Este entorno fue elegido como la plataforma de *back-end* para administrar la información debido a su robustez, confiabilidad y compatibilidad con las tecnologías de *Microsoft* utilizadas en el proyecto. |

Fecha (Semana 2): 08/01/2024 – 12/01-2024

|  |
| --- |
| Durante la segunda semana, se realizó un programa con el propósito de automatizar la generación y distribución de reportes financieros a partir de consultas *SQL* específicas. El procedimiento es ejecutado mediante un script de *Python* que interactúa con una base de datos en *SQL* Server utilizando la biblioteca *pymssql*. Los resultados de las consultas se almacenan en *DataFrames* de la biblioteca *pandas*, que luego son meticulosamente formateados y escritos en un archivo de Excel con múltiples hojas.  En el contexto de este proceso, una característica notable del programa es su capacidad para estandarizar las fechas dentro de los *DataFrames* según diferentes formatos posibles, antes de dedicarse a la tarea de aplicar estilos condicionales. Estos estilos permiten dar formato a las cifras numéricas para que se muestren separadas por puntos en los miles y redondeadas al entero más cercano, además de resaltar los números negativos en color rojo. Para lograr tal resultado, se implementó una combinación de funcionalidades provenientes de *XlsxWriter* y *openpyxl*, ambas bibliotecas avanzadas de *Python* que posibilitan la manipulación de archivos. Se introdujeron formatos personalizados que se asignan dinámicamente a las celdas en base a su contenido, siguiendo reglas específicas y precisas.  El guion establece métodos para el ajuste del ancho de las columnas, asegurando la legibilidad del contenido sin la necesidad de interacción manual. Adicionalmente, contempla la implementación de un ordenamiento en función del primer conjunto de datos numéricos disponible, proporcionando una jerarquización coherente e intuitiva a los ojos del destinatario de los informes.  Por último, se incorporó la lógica para renombrar dinámicamente las hojas del documento con los nombres de las columnas numéricas iniciales, lo que requería una identificación apropiada de estas. Además, el *software* está diseñado para enviar por correo electrónico los reportes generados como archivos adjuntos. En correspondencia con las prácticas modernas de *branding*, se incluyó un logotipo corporativo en el cuerpo del correo usando el logotipo de la compañía, que complementa la presentación profesional del informe.  Esta experiencia implicó una inmersión profunda en la documentación de las librerías ya mencionadas, así como en la ideación de estrategias para resolver desafíos específicos, tales como la estandarización y el formateo condicional de datos. La naturaleza iterativa del desarrollo propició una comprensión integral de las capacidades de las bibliotecas, al tiempo que promovió el desarrollo de una solución robusta y escalable, capaz de servir como base para futuros proyectos de automatización y análisis de datos. |

Fecha (Semana 3): 15/01/2024 – 19/01-2024

|  |
| --- |
| Durante la tercera semana, se llevó a cabo una investigación sobre los diversos protocolos de comunicación utilizados para conectarse a los mercados y bolsas en línea. Se prestó especial atención al protocolo FIX *(Financial Information eXchange)* y su importancia para obtener datos del mercado en tiempo real. Este conocimiento resultó fundamental, ya que el protocolo FIX es ampliamente utilizado en la industria financiera, especialmente en operaciones de trading, para la transmisión de información en tiempo real.  Paralelamente, se creó una documentación detallada en *Notion* que explicaba el funcionamiento del programa existente y cómo se utilizaban los diferentes protocolos de comunicación, incluyendo FIX, para obtener transacciones y datos específicos del mercado. Posteriormente, esta documentación fue enviada a la Fintech Trii para su revisión y se está a la espera de su *feedback*. Este proceso de envío y espera resalta la importancia de una comunicación efectiva y la colaboración con socios externos para el desarrollo de soluciones en el entorno financiero. |

Fecha (Semana 4): 22/01/2024 – 26/01-2024

|  |
| --- |
| Durante la cuarta semana de práctica, se embarcó en la integración de un *WebSocket* con el mercado financiero. La tarea principal fue desarrollar un programa que se conectara al *WebSocket* proporcionado, recibiera mensajes en formato JSON y llevara a cabo acciones específicas basadas en la información recibida. Específicamente en encontrar los precios en línea de las acciones más transadas dentro de la bolsa de Santiago.  Un aspecto destacado de la semana fue la necesidad de reunirse con los clientes para validar y verificar la documentación relacionada con el proyecto. Esta interacción directa con los interesados permitió obtener claridad sobre los requisitos y expectativas, asegurando que el desarrollo se alineara perfectamente con las necesidades del cliente.  Adicionalmente, se dedicó tiempo a mantener una comunicación constante con el encargado del equipo que desarrolló el WebSocket. Esta colaboración fue esencial para comprender en profundidad el funcionamiento interno del WebSocket y asegurarse de que la integración fuera fluida y eficiente.  A la hora de programar, se hizo un código en Python utilizando las librerías `websockets`, `asyncio`, `pandas` y `openpyxl` para gestionar la conexión al WebSocket, realizar solicitudes HTTP para obtener datos adicionales y organizar la información en un archivo de Excel. |

Fecha (Semana 5):29/01/2024 – 02/02/2024

|  |
| --- |
| Durante la quinta semana de práctica, se desplegó en el mundo de Microsoft Azure, un universo lleno de funciones intrigantes y desafíos por descubrir.  En este espacio de trabajo, se crearon dos nuevas funciones de Azure, cada una con su propósito, pero que juntas trabajan con el mismo objetivo.  La primera funcion, *func\_websocket* es una función de tipo HTTP que al gatillarse cumple con las mismas características del *software* desarrollado la semana anterior, con la diferencia de ser una azure *function* y además recibir como input un archivo csv que envía la Fintech Racional todos los días con las órdenes de intención de compra y venta de ese día. Entonces, lo que hace esta función, es buscar todos los nombres de las compañías que fueron transadas ese día y obtener los precios en línea.  La segunda función, *func\_mercados*, también es una función HTTP, que al gatillarse se encarga de estandarizar los datos recibidos desde el archivo de Racional al formato que utilizan internamente para luego ser inyectadas a mercado. Además, esta función destaca en el archivo original todas aquellas órdenes que por algún motivo no pudieron ser procesadas. Como por ejemplo por montos inválidos, cantidad insuficiente, precio punta no encontrado, entre otras.  Por consiguiente, esta semana de práctica fue una experiencia nueva en el mundo de Azure, donde dos funciones se convirtieron en protagonistas de una historia que mezcla lo desconocido a una situación que ocurre día a día en el mercado financiero.  Los desafíos pendientes que quedan es poder desplegar justamente estas funciones en la nube y analizar la escalabilidad y la posibilidad de transformarlo a un microservicio. |

Fecha (Semana 6): 05/01/2024 – 09/02/2024

|  |
| --- |
|  |

Fecha (Semana 7): 12/02/2024 – 16/02/2024

|  |
| --- |
|  |

Fecha (Semana 8): 19/02/2024 – 23/02/2024

|  |
| --- |
|  |

Fecha (Semana 9): 26/02/2024 – 29/02/2024

|  |
| --- |
|  |

|  |  |  |
| --- | --- | --- |
| FIRMA SUPERVISOR  (Fecha) | TIMBRE EMPRESA | **OBSERVACIONES:** |

**OBSERVACIÓN**: Los y las estudiantes que realicen su Práctica II, deben completar y entregar esta bitácora al momento de hacer entrega de su informe. Presentar las copias que sean necesarias, de acuerdo a la duración de la Práctica. Este documento debe tener **nombre, firma y timbre** del supervisor directo del estudiante en la práctica.