

UNIVERSIDAD TECNOLÓGICA NACIONAL
FACULTAD REGIONAL PARANÁ

Informe Práctica Profesional

TUP – UTN FRP

Alumno: **Gastón Chatelet**

Organización: **CONMEGA ACE – StartUp**

Docente: **Ernesto Zapata Icart**

Fecha: **14/12/2023**

Resumen ejecutivo

Este proyecto comprende la realización de una librería de desarrollo creada dentro del equipo de trabajo StartUp perteneciente al grupo de empresas CONMEGA ACE. El periodo de desarrollo de la misma comprende los meses de junio del 2022 a febrero del año 2023 y su implementación y mantenimiento llegan al día de la fecha. La librería, llamada *LitoTable* se enmarca la capa de presentación (front end) de las aplicaciones que lleva a cabo el equipo de trabajo, está basada en material design y pensada para utilizar con el framework AngularJS.

El componente principal de la librería es una tabla, la cual permite agregar o quitar diferentes funcionalidades en la medida en que el proyecto lo requiera. Se buscó adoptar los principios SOLID y el despliegue continuo, puesto que por la modalidad de trabajo del equipo se requiere una constante mejora y la adopción de múltiples funcionalidades.

Índice

Resumen ejecutivo	2
Índice.....	3
Introducción	4
I - Aspectos generales	5
Antecedentes.....	5
Objetivos.....	6
Delimitación de la Práctica Profesional.....	7
Limitaciones.....	8
II – Evaluación institucional	9
Descripción General de la Institución (Empresa).....	9
Objetivos de la Institución	9
Visión y Misión	11
Actividad Institucional (Empresa).....	11
Descripción del grupo humano (con el cuál el alumno trabajó).....	11
Descripción del escenario de trabajo	11
Estructura Organizacional.....	12
III - Desarrollo de la Práctica Profesional.....	13
Descripción de Actividades desarrolladas en la práctica Profesional.....	13
a) Descripción del stack tecnológico:.....	14
b) La necesidad de LitoTable	16
c) Impacto de la <i>LitoTable</i> en el Entorno de Desarrollo.....	18
d) Capacidades técnicas de la LitoTable	19
e) Cronología de la práctica y evolución de la librería.....	20
f) Especificaciones y arquitectura del componente	23
Conclusiones.....	27
Aprendizaje Obtenido de la Realización de la Práctica.....	27
Comentarios Personales del Trabajo Realizado	28
Conclusiones Generales.....	29
Recomendaciones y aportes.....	30
Anexos	31

Introducción

En el dinámico mundo de la tecnología, las oportunidades profesionales a menudo implican la adaptación constante y el desarrollo de nuevas habilidades. En noviembre de 2021, tuve el privilegio de unirme a CONMEGA ACE, un grupo de empresas líderes que incluye nombres reconocidos como Musimundo y Crédito Argentino. Mi viaje en CONMEGA me llevó a una apasionante experiencia en una StartUp dentro de la organización. Como miembro de este equipo, he tenido la oportunidad de participar en el desarrollo de soluciones de software innovadoras que respaldan a diversas empresas del grupo. Comencé como Junior, progresando a Semisenior y finalmente asumiendo el rol de Senior y Líder Técnico del Frontend, supervisando a un equipo de tres profesionales.

Durante mi tiempo en esta StartUp, contribuí al crecimiento y al éxito del equipo y participé en proyectos significativos que demostraron el impacto positivo de mi trabajo. En este trabajo final, compartiré mi experiencia, los desafíos enfrentados, los logros alcanzados y cómo mi contribución profesional ha enriquecido mi desarrollo integral y mi formación académica.

En particular me detendré en el desarrollo de una pieza de software específica: **LitoTable**, ya que encuentro en el desarrollo de esta librería, un nexo tecnológico transversal a la mayoría de los proyectos realizados desde el segundo trimestre del 2022 en adelante por parte del equipo.

I - Aspectos generales

Antecedentes

Mi nombre es Gaston Chatelet. Soy nacido en la ciudad de Santa Fe, pero he vivido en diferentes partes del país hasta radicarme en la ciudad de Paraná, donde vivo actualmente.

Mis primeros pasos en la programación se remontan al año 2017 cuando inicie mi tesis de grado¹ para la Licenciatura en Música Popular para la Universidad Nacional de La Plata, carrera de la que egrese en marzo del 2018 con sobresaliente, siendo mejor promedio de la promoción. Este primer desarrollo fue el de una aplicación web basada en la librería llamada P5js². Dicha aplicación permitía a un usuario interactuar con las diferentes capas de una música en tiempo real.

Luego de este primer acercamiento realice diferentes sistemas, siempre relacionados a lo musical y lo multimedia, como fue el caso de MARTA360³, un software basado en C# para la representación tridimensional a modo de paseo virtual de una casa icónica de la ciudad de La Plata.

Fueron diferentes los proyectos, siempre autogestionados, en los que me involucre y mediante los que fue adquiriendo habilidades dentro del mundo del desarrollo IT. Pero fue recién en el 2020 que tengo mi primer trabajo relacionado a la programación de manera rentada.

¹ <http://sedici.unlp.edu.ar/handle/10915/73597>

² <https://p5js.org/>

³ Repositorio con notas radiales, prensa, etc, <https://drive.google.com/drive/u/4/folders/1DPgcNbg89P-v-IcPibT-t3x8J0kp8L5u>

En este año, ya viviendo en Paraná, fui contratado por el LIMB⁴, a cargo de la Dr. Paola Catalfamo, para realizar una aplicación que permita realizar análisis espacio temporal de la marcha humana. Fue así que desarrolle HAPET⁵, una herramienta que permitiría tomar instantáneas de un clip de video para procesar cuadro a cuadro los distintos momentos de la marcha en los pacientes e inferir así su avance en el tratamiento que pudiesen estar transitando.

Durante la primera parte el 2021 realice proyectos aislados para el extranjero en diferentes plataformas, y a mediados de año ingrese como profesor y dicte un curso de desarrollo full-stack para Digital House⁶. En esta plataforma, dicté el curso para una única cohorte y luego en noviembre del mismo año ingresé a CONMEGA ACE, como analista desarrollador AD2 (equivalente a un programador junior)

Objetivos

- General: El objetivo general de mi Práctica Profesional en CONMEGA ACE fue desarrollar e implementar una solución de software innovadora llamada *LitoTable*. Esta solución tenía como propósito principal modernizar la tecnología legacy utilizada en proyectos anteriores para la representación de datos en tablas en la capa de exposición: UI. Con *LitoTable*, buscamos ofrecer tablas altamente personalizables y funcionales que superaran las limitaciones de las tablas preexistentes, permitiendo a los equipos de desarrollo abordar desafíos más complejos y proporcionar experiencias de usuario mejoradas.

⁴ Laboratorio de Investigación del Movimiento Humano (<http://ingenieria.uner.edu.ar/grupos/limh/>)

⁵ Herramienta de Análisis de Parámetros Espaciales Temporales (<https://github.com/litoralcreativo/hapet>)

⁶ <https://www.digitalhouse.com/>

- Específicos:

1. Diseñar y desarrollar una biblioteca de componentes de tabla altamente personalizables que ofreciera funcionalidades avanzadas, como filtrado, ordenamiento y paginación, para satisfacer las necesidades de múltiples proyectos.
2. Facilitar la integración de *LitoTable* en los proyectos existentes, asegurando la compatibilidad con las tecnologías y sistemas utilizados por CONMEGA ACE.
3. Colaborar con el equipo de desarrollo para identificar y abordar desafíos específicos relacionados con la visualización de datos tabulares, permitiendo una representación efectiva de la información y mejorando la usabilidad de las tablas.
4. Proporcionar documentación completa y recursos de capacitación para los equipos de desarrollo internos, con el fin de garantizar una implementación efectiva y un uso óptimo de *LitoTable*.

Delimitación de la Práctica Profesional

Alcance Temporal: Esta práctica profesional se encuentra intrínsecamente vinculada a mi actividad laboral actual en CONMEGA ACE, pero, a los fines de este trabajo, se delimitará un período de nueve meses, desde junio de 2022 a febrero del 2023. Durante estos meses, se llevó a cabo el desarrollo, implementación y refinamiento de la solución *LitoTable*, integrándola en entornos de trabajo existentes, así como en proyectos nuevos.

Alcance Funcional: La Práctica Profesional se centró exclusivamente en el desarrollo de la librería su integración con los proyectos nuevos y existentes en CONMEGA ACE. No incluyó

otras áreas ni la migración de otras dependencias y librerías legacy, las cuales, si bien se siguen manteniendo, están en un actual proceso de migración.

Alcance Tecnológico: La implementación de *LitoTable* se realizó utilizando tecnologías web y de desarrollo frontend, incluyendo Angular y Material Design. No se abordaron aspectos relacionados con sistemas o tecnologías ajenas a la interfaz de usuario.

Alcance de Aplicación: *LitoTable* se diseñó para ser una solución ampliamente aplicable en múltiples proyectos y aplicaciones dentro de CONMEGA ACE. Sin embargo, su aplicación se limitó al uso interno de la empresa y no se destinó a un público externo.

Alcance Organizacional: La Práctica Profesional se desarrolló dentro de la estructura organizativa de CONMEGA ACE, en colaboración con el equipo de desarrollo StartUp, del cual formo parte. No se implicaron otras organizaciones ni terceros en el proceso, y si bien otros grupos de la empresa a los que brindamos servicio hacen uso de las aplicaciones donde se realizó una implementación de *LitoTable*, estos no promovieron la creación sino nuestro equipo. Actualmente es la StartUp quien mantiene y da soporte a esta librería.

Limitaciones

- El horario de trabajo establecido es de 8hs y las 17hs, de lunes a viernes
- Las horas de trabajo son bajo la modalidad de trabajo remota y se deben fichar las entradas y salidas a través de una aplicación interna provista por la empresa.

II – Evaluación institucional

Descripción General de la Institución (Empresa)

La empresa nace en 1930 a través de Alfonso Valli y su taller de reparación de radios y combinados en la ciudad de Santa Fe. En 1960, con la asociación de sus hijos dan origen a Electrónica Megatone SA, ampliando su actividad al armado de TVs y la comercialización de electrodomésticos. En 1970 abren la primera sucursal en la peatonal San Martín de Santa Fe capital. Para 1980 se convierte en una red nacional de comercialización, y en el 2011 pasa a llamarse Musimundo, adquiriendo la Casa Central ubicada en el ex shopping Paseo del Sol, hoy es una empresa líder en el rubro del retail a nivel nacional con más de 200 sucursales en todo el país. Musimundo es una de las empresas más importantes del grupo de empresas que conforma CONMEGA⁷ ACE, la otra es Crédito Argentino (antes llamada Confina SA).

Objetivos de la Institución

- Honestidad: Fomentar un entorno ético y transparente en todas las operaciones y relaciones de la empresa. Valorar las opiniones y críticas sinceras que aseguren un comportamiento responsable y conforme a las normas internas y externas.
- Involucramiento con los Resultados: Mostrar entusiasmo y pasión por alcanzar los objetivos y compromisos asumidos. Trabajar con la intención de agregar valor, superar desafíos y asumir la responsabilidad de los resultados del trabajo.

⁷ CONMEGA viene de la unión entre CONFina y MEGAtone

- Innovación y Gestión del Cambio: Destacarse por la capacidad de adaptación y eficacia en diversas situaciones. Mostrar una apertura continua a la revisión de operaciones y la capacidad para generar y llevar a cabo ideas creativas y soluciones alternativas.
- Trabajo Colaborativo: Reconocer que para alcanzar aspiraciones, es necesario trabajar trascendiendo barreras sectoriales y geográficas. Estar dispuestos a ser parte de un grupo y cooperar entre sí para lograr objetivos comunes.
- Liderazgo: Los líderes de la empresa son efectivos en la consecución de resultados a través de sus equipos de trabajo. Crean condiciones para que las personas se comprometan y los equipos alcancen sus objetivos. Se caracterizan por acompañar al equipo y estimular la participación, adaptando su estilo de liderazgo a las necesidades individuales.
- Respeto: Promover relaciones interpersonales efectivas basadas en la humildad, empatía y apertura a opiniones y creencias diferentes. Construir un ambiente que valore a los individuos y permita aprovechar la diversidad.
- Foco en el Cliente: Mantener una vocación constante de atención al cliente de manera diferencial. Hacer de la atención al cliente un punto diferenciador de la empresa. Trabajar con actitud de servicio y focalizar esfuerzos en contribuir a la satisfacción del cliente en cualquier ámbito de acción.
- Austeridad: Actuar con plena conciencia y preocupación por el uso eficiente de los recursos disponibles. Contribuir al cuidado del patrimonio de la organización y del medio ambiente en el cual opera.

Visión y Misión

Cada área dentro de las empresas tiene su misión. En lo que respecta a la gerencia de sistemas (de la que formo parte): *Mantener la información consistente, segura y disponible para ser utilizada en todo momento y lugar, agilizando la toma de decisiones y optimizando los procesos del negocio en todos los niveles y ámbitos de la organización, por medio de herramientas tecnológicas de última generación.*

Actividad Institucional (Empresa)

Musimundo es una empresa dedicada a la comercialización de artículos para el entretenimiento y el confort, y la prestación del servicio de post venta dentro del período de garantía.

Descripción del grupo humano (con el cuál el alumno trabajó)

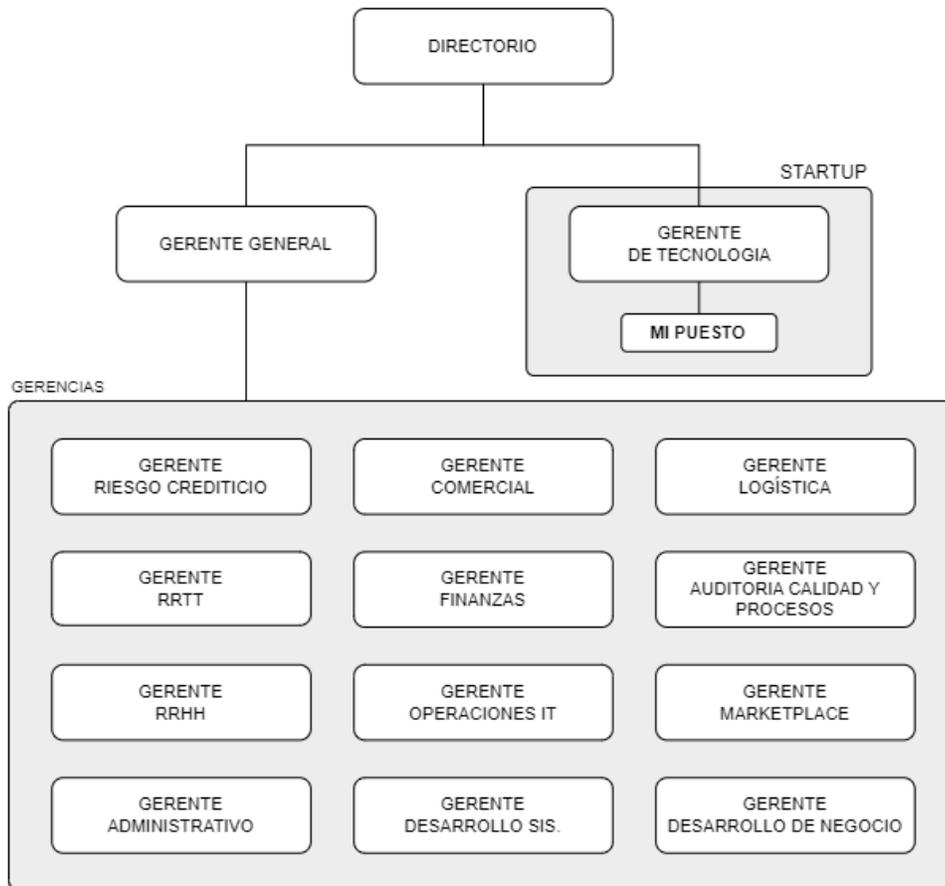
La StartUp se conforma actualmente por 13 personas. El gerente de tecnología como superior, 2 analistas funcionales, 2 analistas QA y 8 analistas desarrolladores de software, entre los cuales me encuentro. De los 8 analistas desarrolladores, yo me desempeño como líder técnico del frontend y tengo a cargo a 3 desarrolladores, y del lado del backend existe un líder técnico con otros tres desarrolladores a su cargo.

Descripción del escenario de trabajo

La StartUp depende directamente del directorio, y el gerente encargado de la misma se encuentra al mismo nivel del gerente general. Es así que la StartUp trabaja paralelamente al resto de gerencias sin depender del gerente general. De igual manera, los servicios que presta el equipo son de las diferentes gerencias, aunque por las características del negocio (retail) la

mayoría de los proyectos priorizados devienen de la gerencia comercial y de la empresa Musimundo.

Estructura Organizacional



III - Desarrollo de la Práctica Profesional

Descripción de Actividades desarrolladas en la práctica Profesional

Antes de adentrarnos en la descripción detallada de las actividades desarrolladas en la práctica profesional, es fundamental comprender el contexto en el que surgió la necesidad de la *LitoTable*. El entorno empresarial de CONMEGA ACE, compuesto por empresas destacadas como Musimundo y Crédito Argentino, requería soluciones tecnológicas de alto calibre para proyectos críticos de administración y gestión de datos. Uno de estos proyectos, conocido como el "Back Office de Marketplace" (BOMKP), marcó el punto de partida para el desarrollo de la *LitoTable*.

En el desafío de ofrecer una herramienta que facilitara la gestión de datos masivos y superara las limitaciones de **Presentation Framework** (*pf-table*)⁸, nació la *LitoTable*. Este componente se convirtió en una respuesta a los obstáculos experimentados en proyectos reales y se forjó con el propósito de proporcionar una solución versátil para la visualización, manipulación y presentación de datos en entornos empresariales complejos.

La *LitoTable* no solo superó las restricciones de su predecesor, sino que también se convirtió en una herramienta valiosa y adaptable que permitió satisfacer las necesidades específicas de una variedad de proyectos. En las próximas páginas, exploraremos en detalle las actividades emprendidas en la creación y evolución de la *LitoTable*, destacando su impacto en el desarrollo de proyectos dentro de CONMEGA ACE

⁸ Herramienta proporcionada por la empresa que actualmente esta siendo sustituida por *LitoTable*

a) Descripción del stack tecnológico:

El desarrollo de la *LitoTable* se encuentra inmerso en un ambiente tecnológico diversificado. En el equipo de desarrollo, trabajamos con una combinación de tecnologías que incluyen **Angular**, **.NET Framework**, e **INFORMIX** como sistema de gestión de bases de datos. Esta combinación de herramientas es esencial para el funcionamiento integral de los proyectos que impulsaron la creación de la *LitoTable* y su posterior adopción en diversos contextos.

- **Angular:**

Angular es un marco de trabajo de código abierto desarrollado y mantenido por Google que permite la creación de aplicaciones web y móviles de alta calidad y rendimiento. En nuestro entorno, Angular se destaca como la principal tecnología para la construcción de aplicaciones de cliente, proporcionando una base sólida para el desarrollo de interfaces de usuario dinámicas y altamente interactivas.

- **.NET Framework:**

En el lado del servidor, utilizamos el entorno de desarrollo .NET Framework, una plataforma de software de Microsoft que permite la construcción, implementación y administración de aplicaciones y servicios web. .NET Framework nos brinda una infraestructura robusta y segura para la lógica de negocio de nuestras aplicaciones, facilitando la comunicación con la base de datos y garantizando un alto nivel de confiabilidad y eficiencia.

- **INFORMIX:**

El sistema de gestión de bases de datos INFORMIX es un componente crítico de nuestro stack tecnológico. Utilizado para el almacenamiento y la recuperación de datos, INFORMIX ofrece una solución robusta y escalable para gestionar grandes conjuntos de información. Su desempeño y confiabilidad se traducen en una administración eficiente de datos para nuestras aplicaciones.

- **Arquitectura:**

El equipo de desarrollo en CONMEGA ACE adopta el enfoque de **Domain-Driven Design** (DDD) como su arquitectura principal. DDD se centra en la resolución de problemas complejos alrededor del dominio del negocio. Esto implica dividir las aplicaciones en módulos y componentes autónomos que reflejan la estructura del negocio.

El enfoque DDD permite una mejor organización de nuestro código y una comprensión más clara de las interacciones entre las diferentes partes de una aplicación. Proporciona un marco para definir y gestionar los límites del dominio y facilita la adaptación a los cambios a medida que evolucionan los requisitos del negocio

Este variado conjunto de tecnologías representa el andamiaje que respalda el desarrollo y funcionamiento de la LitoTable, y es fundamental para comprender cómo se integra esta solución en el contexto de nuestros proyectos.

b) La necesidad de LitoTable

Como se mencionó anteriormente, el proyecto que marcó el inicio de la necesidad de *LitoTable* se denominó "Back Office de Marketplace" (BOMKP). Su propósito principal era proporcionar a los usuarios administrativos la capacidad de gestionar vendedores (sellers) asociados a **megusta**⁹, un incipiente e-commerce que se encontraba en desarrollo por otro sector de la empresa. A medida que los vendedores avanzaban a través de un proceso de carga, validación y verificación, era esencial administrar una gran cantidad de datos de manera eficiente y efectiva.

Sin embargo, la herramienta existente para la representación de datos de forma tabular que hasta entonces se venía utilizando era *pf-table*, un componente de la librería **Presentation Framework**. Esta librería es propia de la empresa, y su mantenimiento se ha dificultado en los últimos años. Esto se traduce a inconsistencias y comportamientos inesperados, los cuales fueron saliendo a la luz en el transcurso de diferentes proyectos en los que fuimos trabajando. Así mismo el componente en particular presentaba una serie de limitaciones que obstaculizaban nuestro progreso, de los cuales podemos nombrar:

- **Ordenamiento inadecuado:** El ordenamiento en *pf-table* se basaba en la representación literal de los datos, lo que llevaba a resultados incoherentes. Por ejemplo, ordenaba "1 - 10 - 2" en lugar de "1 - 2 - 10".

⁹ <https://www.megusta.com/>

- **Limitaciones de estilos y personalización:** *pf-table* carecía de capacidades de personalización de estilos, lo que dificultaba la adhesión a la identidad visual específica de cada proyecto.
- **Falta de documentación adecuada:** La documentación de *pf-table* resultaba insuficiente para comprender sus características y funcionalidades.
- **Paginación del lado del servidor y filtrado complejo:** Implementar la paginación del lado del servidor y configurar el filtrado en *pf-table* era una tarea confusa y poco intuitiva.
- **Formateo de columnas y columnas de acciones:** *pf-table* carecía de capacidades para formatear dinámicamente los datos de las columnas cuando los datos a representar eran complejos a nivel de celda y tampoco permitía la gestión de filas mediante acciones / botones / links.

La conjunción de estos desafíos generó una creciente necesidad de una solución que permitiera gestionar grandes conjuntos de datos de manera efectiva, mejorara la experiencia del usuario y ofreciera un alto grado de personalización. La *LitoTable* surgió como una respuesta directa a esta demanda y rápidamente se convirtió en una herramienta esencial para simplificar la visualización y manipulación de datos en proyectos subsiguientes.

El desarrollo de BOMKP se realizó en el transcurso del 2022. Luego de entregar el proyecto al cliente, el mismo siguió evolucionando y siendo desarrollado por otros equipos del grupo. Hace unos meses el proyecto regreso a nuestro equipo para que ampliemos otras funcionalidades.

c) Impacto de la *LitoTable* en el Entorno de Desarrollo

La *LitoTable* no solo se creó para abordar las limitaciones de las soluciones existentes, sino que también tuvo un impacto significativo en el entorno de desarrollo de la empresa. A lo largo de este punto, exploraremos cómo la *LitoTable* transformó la forma en que se abordan los proyectos y se mejoró la experiencia de desarrollo.

- Impacto en la eficiencia del desarrollo: Antes de la adopción de la *LitoTable*, el desarrollo de tablas de datos en proyectos requería un esfuerzo significativo. La personalización y la implementación de funcionalidades adicionales, como ordenamiento y filtrado, eran tareas que consumían tiempo. Con la *LitoTable*, este proceso se simplificó en gran medida. Configurar y personalizar la tabla se convirtió en una tarea mucho más eficiente en términos de tiempo. La *LitoTable* permitió a los desarrolladores centrarse en aspectos críticos del proyecto en lugar de reescribir funcionalidades estándar.
- Mejora en el rendimiento y la experiencia del usuario: Uno de los cambios más notorios con la introducción de la *LitoTable* fue el aumento en el rendimiento y la experiencia del usuario. La *LitoTable* implementa una forma más eficiente de ordenamiento y filtrado de datos, lo que se traduce en una experiencia más fluida para los usuarios. La velocidad y la capacidad de respuesta mejoradas son especialmente notables cuando se trabaja con conjuntos de datos grandes, lo que era un desafío en las implementaciones anteriores.
- Flexibilidad y Personalización: Un aspecto clave de la *LitoTable* es su capacidad de personalización. Cada proyecto puede tener requisitos y características específicos que deben ser reflejados en la tabla de datos. La *LitoTable* ofrece una interfaz de

configuración que es altamente flexible. Cada atributo de configuración es opcional, lo que permite que se agreguen funcionalidades adicionales de manera sencilla sin afectar las implementaciones anteriores. Esta flexibilidad brinda a los equipos de desarrollo la capacidad de adaptar la tabla a las necesidades de cada proyecto sin esfuerzo adicional.

- Solicitud de ampliación de funcionalidades: La recepción de la *LitoTable* por parte de otros equipos dentro de la empresa fue muy positiva. La versatilidad de este componente permitió que otros equipos solicitaran la adición de nuevas funcionalidades. Algunos de los ejemplos notables de ampliaciones incluyen la implementación de un **color-picker** y la capacidad de estilizar filas dinámicamente según ciertos criterios. Además, se agregaron funcionalidades de selección simple y múltiple de filas, así como botones de acción que emiten eventos con el nombre de la acción y los datos seleccionados.

La *LitoTable* no solo abordó los desafíos técnicos, sino que también mejoró la eficiencia del desarrollo y la experiencia del usuario, consolidándose como una solución sólida para la representación de datos tabulares en proyectos futuros.

d) Capacidades técnicas de la LitoTable

Implementación en Angular: La *LitoTable* está diseñada para trabajar con Angular, el popular framework de desarrollo web. Al estar integrada con Angular, ofrece todas las ventajas y capacidades que Angular proporciona, lo que incluye la administración eficiente de componentes, módulos y rutas. Esta integración con Angular permite una fácil adopción en los proyectos basados en el framework.

Optimización del rendimiento: El rendimiento es un aspecto crítico al trabajar con grandes conjuntos de datos. La *LitoTable* aborda este desafío mediante el uso de conjuntos **Set**

en lugar de **Arrays** para la gestión de datos. La utilización de esta estructura no indexada mejora notablemente la performance, lo que se traduce en una mejora en la velocidad de ordenamiento y filtrado, lo que beneficia directamente la experiencia del usuario. La adopción de prácticas de optimización de rendimiento también contribuye a la eficiencia de la *LitoTable*.

Diseño orientado a interfaces: El diseño de la *LitoTable* sigue una filosofía orientada a interfaces, lo que permite una alta cohesión y bajo acoplamiento. Los componentes pueden comunicarse entre sí de manera efectiva a través de interfaces bien definidas. Esto garantiza una separación clara de responsabilidades y facilita la colaboración en equipos de desarrollo.

e) Cronología de la práctica y evolución de la librería

- **BOMKP**: Como se detalló anteriormente, el proyecto que dio inicio a la *LitoTable* fue el BackOffice de Marketplace, proyecto en el que luego de haber renegado con *pf-table* propuse a mi superior dedicar un tiempo de mi jornada a estudiar la mejor manera de sobrellevar las limitaciones de la tecnología que hasta entonces utilizábamos. En este proyecto, se trabajó en conjunto con la gerencia comercial. La idea era crear una herramienta que permita a los ejecutivos de cuenta realizar la gestión (ABM y Validación) de potenciales vendedores de megusta.com, e-commerce incipiente en el mercado que fue publicado hace pocos meses. El proyecto llevo 4 meses (7 sprints aproximadamente) para su realización. En este proyecto la librería no era una librería como tal, sino que se planteó como un componente más del proyecto. El mismo se creó al igual que se crea cualquier componente en angular, mediante el CLI. Pero fue con el pasar de las semanas que notamos que era insostenible mantener todo el código fuente del componen te dentro de una carpeta, y que su implementación más de un proyecto sería poco amigable. Fue por ello que se optó por crear un proyecto aparte del tipo librería, el cual

únicamente contenga el código fuente de LitoTable. De esta manera, el mantenimiento y la escalabilidad serian sencillos, y la implementación de la librería en cualquier proyecto se resumiría a la instalación de la dependencia mediante el gestor de paquetes **npm**.

Si bien BOMKP maraca el inicio del desarrollo de LitoTable, la librería no fue directamente implementada en este proyecto, puesto que se encontraba en una versión pre productiva, fue con el proyecto de COVFIN en donde si se la incluyo como parte del bunde productivo.

- **COVFIN**: Este proyecto se inició a la par de que el proyecto de BOMKP, aproximadamente cuando aquel proyecto se encontraba en un 70 por ciento. COVFIN es un proyecto que se encarga de la gestión de todas las condiciones de venta y las financiaciones que se ofrecen desde la empresa al momento de la compra de productos. Este proyecto, si bien se inicia desde cero, surge de la necesidad de migrar tecnología legacy. La gestión de las condiciones de venta se realizaba mediante un software 4GL y la gestión de las financiaciones mediante un software CLARION. Para ello se precisó realizar el relevamiento de ambos sistemas a migrar, con los que solo se contaba con el código fuente, y un diagrama de entidad relación que daba cuenta de las tablas afectadas de la base de datos. Este relevamiento conllevó un mes. Luego, cuando se inició con el desarrollo, inmediatamente se adoptó la *LitoTable* como parte de las librerías del front.

En este proyecto la librería se amplió agregando la **columna de acciones**, puesto que se precisaba realizar acciones sobre ciertas filas en específico, también se agregó la posibilidad de seleccionar filas y accionar **botones de acción múltiple**, y sobre el cierre del proyecto se nos pidió y se desarrolló una funcionalidad que permitía la edición de celdas individuales.

Este proyecto conllevó más tiempo, ya que era afectaba una parte crítica del negocio y fueron surgiendo nuevas funcionalidades mientras lo íbamos realizando. Aproximadamente llevo unos 6 meses.

- **GAREXT:** Este proyecto, se inició luego de haber finalizado COVFIN. Este proyecto, el cual continúa hasta la fecha, se basa al igual que COVFIN en la migración de tecnología legacy para la gestión de garantías extendidas. Las garantías extendidas, si bien las provee una empresa externa, son ofrecidas a los clientes al momento de realizar una compra. Si el cliente compra esta garantía extendida, nuestra empresa hace las veces de intermediario con la aseguradora. La gestión de estas garantías son uno de los principales ingresos que tiene la empresa y se realizaban mediante sistemas 4GL y Clarion. Por lo que lo aprendido en COVFIN nos sirvió de marco para abordar el relevamiento de esta nueva migración.

En este proyecto, se desarrolló con mayor medida la **paginación del lado del servidor**. Esto significó refactorizar parte del código, puesto que nos e había considerado previamente la necesidad de este tipo de comportamiento. Se aprovechó esta refactorización para mejorar la forma de entrada de configuraciones de la tabla, así como también el diálogo entre componentes internos. Particularmente se planteó que para el uso de un tipo u otro de paginación (CSP o SSP)¹⁰ la clase (componente) que contuviese a la tabla debería implementar una interfaz u otra. Estas interfaces son genéricas y toman como genérico a una la clase (DTO) que se representara en las columnas de la tabla: `LitoSSP<T>` y `LitoCSP<T>`.

¹⁰ CSP: Client Side Pagination. SSP: Server Side Pagination

f) Especificaciones y arquitectura del componente

Decoradores y definición de columnas: Si bien la librería y el componente se iniciaron desde cero, se tuvieron en cuenta algunas de las características y cierta arquitectura de su predecesor *pf-table*. Puntualmente lo que se tuvo en consideración fue el modo de definición de las columnas de la tabla mediante **decoradores**¹¹. En el contexto de Angular, los decoradores son una característica esencial que se utiliza para extender y modificar el comportamiento de las clases, componentes, servicios y otros elementos de Angular. Los decoradores son una parte fundamental de la metaprogramación en Angular, lo que significa que permiten agregar metadatos y configuración a las clases de una manera declarativa y legible. En lo que respecta a la *LitoTable*, la definición de las columnas se hace mediante el decoramiento de atributos de una clase con la función `TableColumn`, en donde se especifican las características mediante una interfaz llamada `TableColumnMetadata`. Cuando pasamos al componente la clase decorada, este puede crear las columnas tomando sus características, entre las cuales se encuentran el formato del tipo de dato (texto, numérico, currency, cuit, booleano, icono, etc), alineación del encabezado, alineación del contenido, visibilidad, orden, nombre de la columna, modo de crecimiento horizontal, tooltip del encabezado, grupo de pertenencia, carga asincrónica, etc. A continuación, se muestra como se decora el atributo de una clase que se necesita representar como columna en el componente *LitoTable*:

¹¹ <https://refactoring.guru/es/design-patterns/decorator>

```
export class Empleado {
  @TableColumn({
    columnName: 'Nombre',
    order: 1,
    contentAlign: ColumnContentAlignment.RIGHT,
    type:ColumnType.STRING,
    nonDeselectable: true
  })
  name: string

  @TableColumn({
    columnName: 'Sueldo',
    order: 2,
    contentAlign: ColumnContentAlignment.RIGHT,
    type:ColumnType.FLOAT,
    format: "1.2",
    headerTooltip: "Dinero que percibe el empleado",
  })
  sueldoPercibido: number
}
```

El bloque anterior define la clase Empleado, la cual cuenta con dos atributos de clase: name y sueldoPercibido. Estos atributos fueron decorados de tal manera que serán representados mediante la LitoTable con una alineación a la derecha, tendrán los encabezados Nombre y Sueldo, uno será representado como texto y el otro como numérico de punto flotante con dos dígitos luego de la coma, además uno de ellos tendrá un tooltip cuando se posicione el mouse sobre el encabezado.

Este modo de definir las columnas de la tabla no sufrió mayores modificaciones en las sucesivas versiones que se desarrollaron, mas allá de la ampliación de la interfaz TableColumnMetadata. Ahora bien, en el pasaje de COVFIN a GAREXT hubo un cambio notorio en lo que respecta a la definición de cómo implementar la tabla dentro de un componente padre.

Implementación del componente: Si bien anteriormente describimos el modo de definición del DTO a representar en la tabla. Para la implementación de la misma debemos llevar a cabo un conjunto de pasos. Estos pasos se vieron notablemente modificados en el pasaje de un proyecto a otro. La versión implementada en COVFIN solo precisaba que se pasen por Input el tipo (una instancia de la clase decorada) y un observable del listado de elementos a representar (muchas veces el resultado de la consulta http). Cuando trabajamos en el proyecto GAREXT se decidió refactorizar ciertas características de la *LitoTable*, entre las cuales se encontraba la definición de una interfaz genérica que reduce el grado de cometer errores al momento de implementar el componente. A continuación, detallamos las dos interfaces planteadas:

```
export interface LitoCSP<T> {
    tableConfig: TableConfigurations;
    type: T;
    lito: LitotableComponent<T>;
    tableConfiguration(): void;
    tableSourceFetching(): void;
    inputSource: Observable<T[]>;
}

export interface LitoSSP<T> {
    tableConfig: TableConfigurations;
    type: T;
    lito: LitotableComponent<T>;
    tableConfiguration(): void;
    tableSourceFetching(): void;
    inputSource: Observable<LitotableSSPResponse<T>>;
    SSPPaginatorOrSortChange($event: {
        paginator: MatPaginator;
        sort: MatSort;
    }): void;
}
```

En el bloque anterior podemos notar que ambas interfaces comparten miembros, los cuales son necesarios implementar siempre. Ahora bien, en lo que respecta al miembro

`inputSource`, si bien ambas interfaces lo definen, el tipo del mismo difiere. En la paginación del lado del cliente (CSP), este miembro espera un observable genérico de la lista de elementos a representar, en la paginación del lado del servidor (SSP), este miembro espera un observable de un objeto genérico: `LitotableSSPResponse<T>`. Esta clase añade datos propios de la paginación del lado del servidor: número de página, elementos por página, y total de elementos.

Además, la interfaz SSP define un método muy necesario para este tipo de paginación: `SSPPaginatorOrSortChange`. Mediante este método, se podrá escuchar cualquier cambio que exista en la tabla respecto al número de página o el ordenamiento de alguna de sus columnas. La tabla emitirá un evento cuando se realice un cambio en alguna de estas estructuras y se podrá capturar mediante este método. A continuación, se muestra como se implementa la `LitoTable` en el template HTML.

```
<lito-table
  [type]="type"
  [inputSource]="inputSource"
  [configurations]="tableConfig"
  (exportOutput)="onExportOutput()"
  (singleActionOutput)="onSingleActionOutput($event)"
  (SSPPaginatorOrSortChangeEmmitter)="SSPPaginatorOrSortChange($event)"
  (sourceFetched)="onSourceFetched($event)"
></lito-table>
```

Aquellos miembros definidos entre corchetes (`type`, `inputSource`, `configurations`) son los valores de entrada del componente. Allí pasamos los datos a representar, la configuración de la tabla y la instancia de la clase (DTO) para la definición de columnas. Los miembros definidos entre paréntesis son eventos de salida del componente. Por ejemplo, si notamos la línea:

```
(SSPPaginatorOrSortChangeEmmitter)="SSPPaginatorOrSortChange($event)"
```

Podemos notar que al emitirse el evento `SSPPaginatorOrSortChangeEmmitter`, este es capturado en el método `SSPPaginatorOrSortChange($event)` de la clase contenedora, donde `$event` es el payload que lleva los datos del nuevo estado de la tabla, aquí vendrán datos de número de página, número de elementos por página, columna a ordenar y dirección del ordenamiento en caso de existir.

Conclusiones

En este apartado, se presentan las conclusiones derivadas de la realización de la práctica profesional con un enfoque en el desarrollo y la implementación de la `LitoTable`. Estas conclusiones abarcan tanto los aspectos específicos relacionados con la experiencia con la `LitoTable` como los aprendizajes generales que se extrajeron durante el proceso.

Aprendizaje Obtenido de la Realización de la Práctica

La experiencia adquirida durante la práctica profesional ha sido fundamental para el crecimiento y desarrollo profesional. El aprendizaje abarca una amplia gama de áreas, desde aspectos técnicos hasta habilidades interpersonales. Se destacan los siguientes puntos clave:

- Dominio de Angular y Material Design: A lo largo de la práctica, se profundizó en el conocimiento y la comprensión de Angular y Material Design. Esto incluye la implementación efectiva de componentes, la optimización de rendimiento y la gestión de estados.

- Métodos Ágiles de Desarrollo: La práctica profesional permitió una inmersión en métodos ágiles de desarrollo, lo que resultó en una mayor eficiencia y colaboración en el equipo.
- Resolución creativa de problemas: Se adquirió la capacidad de abordar problemas técnicos y desafíos de diseño de manera creativa y efectiva, encontrando soluciones innovadoras.
- Gestión de proyectos: Se obtuvo experiencia en la planificación y gestión de proyectos de desarrollo de software, lo que incluye la asignación de recursos y la gestión del tiempo, así como también una participación activa en las etapas de relevamiento.
- Flexibilidad y adaptabilidad: Se destacó la importancia de desarrollar sistemas flexibles y adaptables. La LitoTable se diseñó con la capacidad de extender sus funcionalidades de manera sencilla, lo que se tradujo en la capacidad de adaptarse a las necesidades cambiantes de diferentes proyectos.

Comentarios Personales del Trabajo Realizado

La experiencia de llevar a cabo la práctica profesional y, en particular, trabajar en el desarrollo e implementación de la LitoTable ha sido un período de crecimiento y desafío profesional significativo. Durante este tiempo, me he encontrado obstáculos y he superado desafíos, lo que me ha llevado a un valioso aprendizaje y desarrollo personal.

Uno de los aspectos más notables de este trabajo ha sido la oportunidad de abordar problemas técnicos complejos y encontrar soluciones creativas. En particular, la necesidad de crear una tabla flexible y personalizable que pudiera integrarse de manera efectiva en varios proyectos fue un desafío emocionante. La LitoTable no solo cumplió con éxito esta función,

sino que también demostró ser una herramienta versátil que puede adaptarse a una amplia gama de necesidades de desarrollo.

El dominio adquirido de Angular y Material Design ha sido fundamental para el éxito de este proyecto. La comprensión profunda de estos marcos permitió una implementación eficiente y un rendimiento óptimo. Además, la práctica profesional proporcionó la oportunidad de adentrarme en el mundo de los métodos ágiles de desarrollo, lo que mejoró la colaboración y la eficiencia en el equipo.

En retrospectiva, me siento satisfecho con los resultados obtenidos, pero siempre veo margen para la mejora. El proceso de desarrollo de la *LitoTable* también me ha enseñado la importancia de mantener la documentación actualizada y accesible para todo el equipo. La falta de una documentación completa en Presentation Framework nos recordó la importancia de proporcionar recursos claros y comprensibles.

Este período de práctica profesional ha fortalecido mi pasión por la programación y la resolución creativa de problemas. Me ha dejado ansioso por asumir nuevos proyectos y enfrentar desafíos futuros. En resumen, esta experiencia ha sido un valioso capítulo en mi desarrollo profesional, y espero aplicar los conocimientos y habilidades adquiridos en futuros proyectos.

Conclusiones Generales

El enfoque en la creación de una tabla modular, personalizable y escalable ha demostrado ser acertado. La flexibilidad de la *LitoTable* ha permitido adaptarse a una amplia gama de proyectos y requisitos sin requerir una reescritura significativa. Esto ha sido especialmente valioso para proyectos en los que se necesitaba una interfaz de usuario rica en funcionalidades de tablas.

La capacidad de configuración de la *LitoTable*, que permite a los desarrolladores ajustar sus propiedades y comportamientos, ha sido bien recibida. Esto ha contribuido a una experiencia de usuario más agradable y a la alineación con la identidad visual y los requisitos específicos de cada proyecto.

Recomendaciones y aportes

- Continuar el desarrollo de la *LitoTable*: La librería debe mantenerse actualizada para seguir siendo compatible con las últimas versiones de Angular y Material Design. Se recomienda asignar recursos para el mantenimiento continuo, lo que incluye la actualización de dependencias y la resolución de problemas o vulnerabilidades de seguridad que puedan surgir.
- Estrategias de Pruebas Sólidas: A medida que la *LitoTable* se convierte en una parte integral de los proyectos, es fundamental establecer estrategias de pruebas sólidas. Se recomienda implementar pruebas unitarias y de integración para garantizar el correcto funcionamiento y la estabilidad de la herramienta en diferentes escenarios.
- Documentación y Capacitación: Continuar manteniendo y mejorando la documentación es esencial. Además, se recomienda proporcionar capacitación a los nuevos miembros del equipo o desarrolladores que no estén familiarizados con la *LitoTable*. Esto acelerará su adopción y asegurará un uso efectivo.

Anexos

	Código	Descripción	Alta Programada	Fecha de ejecución	Creación	Fecha de baja	Estado
<input type="checkbox"/>	33	test1Noejecuta2			02/11/2023	02/11/2023	Anulada
<input type="checkbox"/>	32	test1Ok	02/11/2023	02/11/2023	02/11/2023		Ok
<input type="checkbox"/>	31	test1NoEjecuta	02/11/2023	02/11/2023	02/11/2023		Error
<input type="checkbox"/>	30	test1falla	02/11/2023	02/11/2023	02/11/2023		Ok
<input type="checkbox"/>	29	dfs			01/11/2023		Listo

Ilustración 1 Una de las implementaciones de LitoTable en el proyecto COVFIN. A) Uso de *mat-chips* en la columna Estado. B) Coloración grisácea de una fila entera para denotar invalidez o anulación. C) En la parte superior un conjunto de botones también seteados en la interfaz de configuración de la tabla. D) Filtrador general con las columnas a filtrar y un segundo filtro más avanzado, ambos vacíos.

	Código	Descripción	Cantidad de detalles	Fecha Alta	Fecha Baja	Fecha Actualización
<input type="checkbox"/>	1.091	TB 1, 3 Y 6 S/I	3	03/06/2013		18/08/2022
<input checked="" type="checkbox"/>	9.920	GC - 1 - 3 Y 6C SI	3	10/05/2021		23/08/2021

Ilustración 2 Otra implementación de la tabla para COVFIN. A) filtro utilizado para las columnas código y descripción, lo que hizo que de 744 elementos activos, queden dos. B) Segundo filtro (activos), lo que hace que se muestren sean solo 744 de 1250. C) fila seleccionada, lo que hace que se habiliten ciertos botones superiores (consultar, editar, duplicar, baja, historial) y se deshabilite el botón de nueva financiación.

UTN - FACULTAD REGIONAL PARANÁ
TECNICATURA UNIVERSITARIA EN PROGRAMACIÓN
INFORME PRACTICA PROFESIONAL

Comprobante							
Tipo	Punto vta.	Número	Local	Certificado ↑	Observación		Acciones
8	1885	977	255	241	[Desc.Prod.] PM=CEL.MOTO G42 XT2233-1 AZUL		🔍 🖨️ ↺
6	5796	228	288	241	[Desc.Prod.] PM=CEL.MOTO G42 XT2233-1 AZUL		🔍 🖨️ ↺
6	5796	256	288	269	[Desc.Prod.] PM=CEL.MOTO G42 XT2233-1 AZUL		🔍 🖨️ ↺
6	5796	282	288	285	[Marca] PM=BLACK POINT / Cardif=BLACKPOINT		🔍 🖨️ ↺ ☰
6	5796	305	288	302	[Marca] PM=BLACK POINT / Cardif=BLACKPOINT		🔍 🖨️ ↺ ☰

Items por página 5 1 - 5 de 1214 < >

Ilustración 3 Se puede apreciar una tabla con columnas y un agrupador de columnas (Comprobante). Dos de las filas se encuentra diferente al resto por reglas de estilo asociadas a uno de los atributos del DTO. Se aprecia también la columna de acciones con 3 – 4 acciones específicas para cada fila. Estas acciones emiten un evento al componente contenedor. También se observa que existe ordenamiento ascendente en la columna Certificado. La paginación es del lado del cliente, existiendo 1214 elementos, de los cuales solo se están mostrando del 1 al 5

Comprobante							Certificado			Meses de gara	
Fecha de cte.	Tipo	Punto vta.	Número cte.	Posdatada	Factura original	Tipo y nro. doc.	Local	Número	Conciliado	Provee	Acciones
31/01/2023	6	5431	1543	NO	0-0-0	96 - 17735870	90	56141	P	12	🔍
31/01/2023	6	5431	1543	NO	0-0-0	96 - 17735870	90	56142	P	12	🔍
31/01/2023	6	254	95062	NO	0-0-0	96 - 13776750	57	55756	P	12	🔍
31/01/2023	6	1878	14601	NO	0-0-0	97 - 16404739	252	12952	P	12	🔍
31/01/2023	6	790	40414	NO	0-0-0	98 - 8009920	131	45590	P	12	🔍

Items por página 5 11 - 15 de 63810 < >

Ilustración 4 Ejemplo con paginación del lado del servidor en GARTEXT. En este caso, solo se tiene en memoria 5 elementos, y no 63810. El payload de esta petición indico que se necesitaba la pagina 2 con límite de 5. La respuesta del servicio (API) fue un listado de 5 elementos con los datos de paginación más el total de elementos.