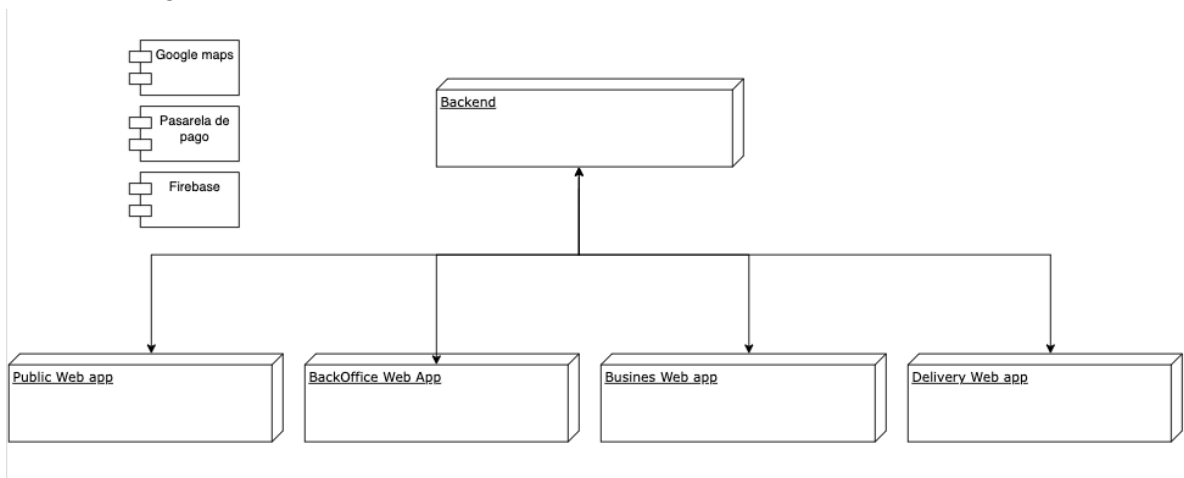


Spike/Bain Challenge

1. Las piezas de software serían las siguientes:
 - a. Aplicación backend: Albergará la lógica de negocio y conectará con la base de datos, proveerá de información a las 4 aplicaciones frontend que se desarrollaran.
 - b. Aplicación web pública: Consumirá los listados y negocios desde el backend y usará los servicios de terceros para autenticación, pagos y mapas
 - c. Aplicación web backoffice: Registro de los negocios creados y cartas. Se conecta con backend
 - d. Aplicación web cliente negocio: Permitirá a los negocios recibir pedidos
 - e. Aplicación web repartidor: Recibirá los pedidos y actualiza el status del pedido
 - f. Diagrama



2. La arquitectura es cliente servidor: La elegí porque es sencilla de implementar, provee mejor escalabilidad de infraestructura y de desarrollo que una monolítica y permite que cada aplicación se pueda concentrar en su público objetivo
3. Usaría kanban pero modificado con el rol de product owner de scrum que lo ocuparía la/el impact manager. El rol de product owner definiría el roadmap del producto y evaluará la complejidad con ayuda de mi para la complejidad de software y de los Data Scientist y Machine learning engineer para plantear una eficiente captación de datos para su futuro consumo.
Solo mantendrá la ceremonia de sprint review para que la product owner pueda inspeccionar el avance y tener mejor información para la toma de decisiones del producto. El resto de reuniones serían ad hoc.
El equipo al ser chico y querer entregar un producto útil pronto debe priorizar eficiencia y con esta metodología en mi experiencia funciona bien con el contexto brindado

4. El workflow que usaría para el manejo de git seria con 2 ramas principales:
 - a. Master
 - b. Dev

Los feature branch irían con un pull request a master para hacer un code review, pero se les haría merge directo a Dev para poder probar y asegurar que todo funciona de manera correcta. Si una feature se prueba con éxito en Dev recién se aprobará el PR a master. Periódicamente la rama Dev se reseteara a la rama Master porque la idea es que Dev tenga el mismo contenido que Master pero al ser Master la rama de despliegue lo que llegue ahí debe ser desplegable.

5. Durante el desarrollo no creo que sea necesario agregar un integrante extra al equipo, ya que ya se tiene lo suficiente para crear un prototipo funcional de manera eficiente. Sin embargo, después que el prototipo funcione bien agregaria los siguientes roles según lo especificado:
 - a. Full stack developer: Para acelerar el desarrollo de los componentes. Ingresar apenas se valide el prototipo y se replantee el roadmap
 - b. QA: Para poder delegar la responsabilidad de pruebas. Ingresar cuando el Full Stack developer ya tenga 1 o 2 meses y el ritmo de desarrollo este fluido
 - c. DevOps: Para volver al desarrollo más fluido y permitir la entrega más eficiente de nuevos features. Ingresar junto al QA
 - d. Data Engineer o Data Science: Cuando el producto esté en el product/market fit para trabajar la data recolectada e implementar formas buenas de que Product pueda evaluar la información y tomar decisiones
6. Priorizar que los desarrolladores tengan conocimiento del problema/necesidad que resuelve cada feature y priorizar resolver esto en cada tarea que se realiza. Eventualmente girar a prácticas como las recomendadas por Shape Up para poder ser un negocio eficiente y que los deadlines sean fijos pero el alcance total de la tarea sea variable en la medida que solucione el problema o necesidad.