

Grupo 1: Proyecto "Sistema de Control de Calidad para Fabricación Automotriz"

1. Modelo de Desarrollo más Apropriado: Modelo en Cascada (Waterfall)

2. Justificación de la Elección:

- **Requisitos Estables:** Los requisitos son extremadamente estrictos, bien definidos desde el inicio y con normativa innegociable, lo cual se alinea perfectamente con la naturaleza secuencial y de requisitos fijos del Cascada.
- **Énfasis en Precisión:** El modelo Cascada, al exigir la finalización y aprobación de cada fase antes de avanzar, asegura una alta precisión y cumplimiento de especificaciones, crucial para un sistema crítico de calidad.
- **Documentación Rigurosa:** Este modelo promueve una documentación exhaustiva en cada etapa, lo cual es vital para la trazabilidad y auditoría en un sector tan regulado como el automotriz.

3. Riesgos Potenciales y Mitigación:

- **Riesgo 1: Descubrimiento Tardío de Errores:** Si un error en los requisitos o el diseño se detecta tarde, corregirlo es muy costoso.
 - **Mitigación:** Realizar revisiones formales y exhaustivas (walkthroughs y inspections) en cada fase, con participación de expertos en calidad y usuarios clave, antes de dar por terminada la fase.
- **Riesgo 2: Falta de Flexibilidad ante Cambios Menores:** Aunque los requisitos son estables, podrían surgir pequeñas aclaraciones que el modelo Cascada maneja con dificultad.
 - **Mitigación:** Establecer un proceso formal de gestión de cambios que evalúe rigurosamente el impacto y el costo de cualquier modificación propuesta.

4. Modelo Alternativo Descartado: Modelo Ágil

- **Argumento 1:** La alta flexibilidad y adaptación a cambios del modelo Ágil choca con la necesidad de requisitos estables y la rigidez normativa de este proyecto. Un flujo constante de cambios podría comprometer la consistencia y precisión requeridas.
 - **Argumento 2:** La menor documentación formal del Ágil sería insuficiente para la trazabilidad y auditoría que un sistema de control de calidad automotriz exige, donde cada decisión y especificación debe estar detalladamente registrada.
-

Grupo 2: Proyecto "Aplicación Móvil de Recetas Saludables Personalizadas"

1. Modelo de Desarrollo más Apropriado: Modelo Incremental (o una aproximación Ágil)

2. Justificación de la Elección:

- **Evolución del Producto:** La necesidad de añadir funcionalidades avanzadas en futuras versiones, dependiendo de la retroalimentación del mercado, se alinea perfectamente con la entrega por módulos o incrementos del software.
- **Adaptación a la Retroalimentación:** Permite lanzar una versión básica (MVP), recopilar feedback de los usuarios y luego iterar y mejorar el producto en incrementos posteriores, lo cual es crucial para una startup que depende de la aceptación del mercado.
- **Tiempo de Mercado Rápido:** Al entregar versiones funcionales de forma progresiva, la startup puede obtener una presencia en el mercado más rápidamente con las funciones básicas, atrayendo usuarios e inversores.

3. Riesgos Potenciales y Mitigación:

- **Riesgo 1: Problemas de Integración:** Si los módulos o incrementos no están bien planificados, la integración de nuevas funcionalidades podría ser compleja y generar errores.
 - **Mitigación:** Establecer una arquitectura modular sólida desde el principio y realizar pruebas de integración continuas después de cada incremento.
- **Riesgo 2: Desviación del Alcance (Scope Creep):** La flexibilidad de añadir características puede llevar a la constante adición de funcionalidades, extendiendo el proyecto indefinidamente.
 - **Mitigación:** Definir claramente el alcance de cada incremento o versión, priorizando las funcionalidades con mayor valor de negocio y manteniendo un backlog de características.

4. Modelo Alternativo Descartado: Modelo en Cascada (Waterfall)

- **Argumento 1:** El Cascada no es adecuado para un producto que se espera que evolucione y cambie según la retroalimentación del mercado. La rigidez para manejar cambios tardíos sería un obstáculo para una startup.
 - **Argumento 2:** La entrega tardía de un producto completamente funcional en Cascada no permitiría a la startup obtener retroalimentación temprana ni adaptarse rápidamente a las demandas del mercado.
-

Grupo 3: Proyecto "Software de Gestión de Tráfico Aéreo de Nueva Generación"

1. Modelo de Desarrollo más Apropriado: Modelo en Espiral (Spiral Model)

2. Justificación de la Elección:

- **Gestión de Riesgos Críticos:** Este es un proyecto de alto riesgo (consecuencias catastróficas). El modelo Espiral integra el análisis de riesgos en cada ciclo, permitiendo identificar, evaluar y mitigar proactivamente los peligros en etapas tempranas.
- **Complejidad del Sistema:** La naturaleza de subsistemas interconectados y la alta complejidad se benefician de las iteraciones del Espiral, que permiten abordar la complejidad en etapas y refinar las soluciones.
- **Requisitos Evolutivos (por riesgos):** Aunque los requisitos son críticos, el Espiral permite su refinamiento a medida que se entienden mejor los riesgos y las soluciones técnicas, asegurando que el producto final sea seguro y efectivo.

3. Riesgos Potenciales y Mitigación:

- **Riesgo 1: Costo y Duración Elevados:** La naturaleza iterativa y el énfasis en el análisis de riesgos pueden hacer que el proyecto sea costoso y prolongado.
 - **Mitigación:** Establecer métricas claras para el progreso en cada ciclo y realizar revisiones exhaustivas para asegurar que se está avanzando de manera eficiente y que los riesgos son manejables.
- **Riesgo 2: Demanda de Expertos en Riesgos:** Requiere personal con experiencia en gestión de riesgos de alto nivel.
 - **Mitigación:** Formar un equipo de gestión de riesgos dedicado y capacitar al personal clave en metodologías de análisis y mitigación de riesgos.

4. Modelo Alternativo Descartado: Modelo de Prototipos

- **Argumento 1:** Aunque el prototipado puede ayudar a entender requisitos, la naturaleza crítica y de seguridad de un sistema de tráfico aéreo no puede basarse en un prototipo "rápido y desechable" sin una evaluación de riesgos rigurosa, que no es el foco principal del Prototipos.
- **Argumento 2:** Los prototipos pueden generar falsas expectativas o llevar a soluciones que, aunque visualmente atractivas, no cumplan con las estrictas normativas de seguridad y rendimiento que este sistema exige.

Grupo 4: Proyecto "Plataforma Educativa Online para Cursos Cortos"

1. Modelo de Desarrollo más Apropriado: Modelo de Prototipos

2. Justificación de la Elección:

- **Validación de Interfaz/Funcionalidad:** La necesidad de probar ideas de diseño y flujos de usuario con estudiantes y profesores antes del desarrollo completo es la esencia del prototipado.
- **Clarificación de Requisitos:** Ayuda a la universidad a definir mejor sus requisitos detallados y a visualizar el sistema, reduciendo la incertidumbre y asegurando que el producto final satisfaga las expectativas del usuario.
- **Mejora la Comunicación:** Facilita la comunicación entre el equipo de desarrollo y los stakeholders (universidad, estudiantes, profesores) al proporcionar algo tangible para probar y discutir.

3. Riesgos Potenciales y Mitigación:

- **Riesgo 1: Expectativas Irreales:** Los usuarios pueden confundir el prototipo con el producto final y esperar que el prototipo sea completamente funcional y robusto.
 - **Mitigación:** Comunicar claramente que el prototipo es una versión preliminar para obtener retroalimentación y gestionar las expectativas sobre sus limitaciones.
- **Riesgo 2: Prototipo Mal Diseñado:** Si el prototipo no se gestiona bien, puede convertirse en un "prototipo desechable" que no aporta valor o se convierte en la base del producto final, heredando sus defectos.
 - **Mitigación:** Establecer objetivos claros para cada prototipo, definir qué se espera aprender de él y cuándo se considera que ha cumplido su propósito antes de pasar a una fase de desarrollo más formal.

4. Modelo Alternativo Descartado: Modelo en Cascada (Waterfall)

- **Argumento 1:** Con requisitos poco claros en las etapas iniciales y la necesidad de probar la interfaz y la experiencia del usuario, el modelo Cascada, que exige requisitos fijos al inicio, sería inflexible y llevaría a un producto final que podría no satisfacer a los usuarios.
- **Argumento 2:** La falta de retroalimentación temprana del usuario en Cascada significaría que los errores de diseño o funcionalidad solo se detectarían en las fases de pruebas o incluso después de la implementación, con un alto costo de corrección.

Grupo 5: Proyecto "Sistema de Facturación y Contabilidad para Pequeñas Empresas"

1. **Modelo de Desarrollo más Apropiado: Modelo Incremental (o Ágil con Sprints bien definidos)**

2. Justificación de la Elección:

- **Entrega Progresiva de Valor:** Permite lanzar una versión con funcionalidades básicas de contabilidad y luego añadir módulos o mejoras en versiones posteriores, respondiendo a las necesidades del mercado y regulaciones.
- **Gestión de un Equipo Pequeño:** Un equipo pequeño puede enfocarse en entregas manejables y de calidad en cada incremento, en lugar de intentar construir todo de una vez.
- **Adaptación a Regulaciones:** Si las regulaciones fiscales cambian, se pueden incorporar en un incremento futuro sin tener que rediseñar todo el sistema.

3. Riesgos Potenciales y Mitigación:

- **Riesgo 1: Problemas de Arquitectura:** Si la arquitectura inicial no es lo suficientemente flexible, la adición de nuevos incrementos puede volverse compleja o inestable.
 - **Mitigación:** Invertir tiempo en un diseño arquitectónico robusto y escalable al inicio del proyecto.
- **Riesgo 2: Consistencia entre Incrementos:** Asegurar que cada incremento mantenga la coherencia con los anteriores y que la experiencia del usuario sea fluida.
 - **Mitigación:** Mantener una guía de estilo y estándares de desarrollo estrictos, y realizar pruebas de regresión exhaustivas en cada nueva versión.

4. Modelo Alternativo Descartado: Modelo en Cascada (Waterfall)

- **Argumento 1:** Aunque las funciones principales son conocidas, la expectativa de incorporar nuevas características y mejoras periódicamente choca con la poca flexibilidad del Cascada para manejar cambios después de la fase de requisitos.
- **Argumento 2:** La entrega de un producto completo y final en Cascada retrasa el retorno de la inversión y la obtención de retroalimentación de los usuarios finales, lo cual es importante para un producto de mercado.

Grupo 6: Proyecto "Videojuego Indie Multijugador"

1. **Modelo de Desarrollo más Apropiado: Modelo Ágil (Scrum o XP)**
2. **Justificación de la Elección:**

- **Requisitos Altamente Cambiantes:** Las mecánicas de juego, equilibrio y monetización se refinarán con la retroalimentación de la comunidad, lo cual requiere la alta flexibilidad y capacidad de adaptación del Ágil.
- **Colaboración Constante:** El Ágil promueve la colaboración diaria entre diseñadores, programadores y artistas, crucial para un videojuego donde el arte y la jugabilidad deben integrarse a la perfección.
- **Retroalimentación Continua:** Las entregas frecuentes de software funcional (sprints) permiten obtener retroalimentación temprana de los testers, probar ideas y reaccionar rápidamente a los problemas o nuevas ideas.

3. Riesgos Potenciales y Mitigación:

- **Riesgo 1: Desviación del Alcance (Scope Creep):** La flexibilidad extrema puede llevar a añadir demasiadas funciones, extendiendo el desarrollo.
 - **Mitigación:** Mantener un Product Backlog priorizado y bien gestionado, con sprints de duración fija, y ser rigurosos en la definición de "hecho" para cada sprint.
- **Riesgo 2: Falta de Documentación Detallada:** El enfoque del Ágil en "software funcionando sobre documentación exhaustiva" podría dejar detalles críticos del diseño o la arquitectura sin registrar.
 - **Mitigación:** Implementar prácticas de "documentación útil y mínima necesaria" (por ejemplo, diagramas de arquitectura actualizados, comentarios de código claros) y reuniones regulares de intercambio de conocimiento.

4. Modelo Alternativo Descartado: Modelo en Cascada (Waterfall)

- **Argumento 1:** La naturaleza altamente cambiante e incierta de los requisitos de un videojuego indie es incompatible con la rigidez y la necesidad de requisitos fijos del Cascada.
- **Argumento 2:** La entrega única al final del Cascada no permitiría la iteración rápida, la prueba de mecánicas de juego ni la adaptación a la retroalimentación crucial de la comunidad, lo que es vital para el éxito de un videojuego.

Grupo 7: Proyecto "Migración de un Sistema Legado de Recursos Humanos"

1. **Modelo de Desarrollo más Apropriado: Modelo en Cascada (Waterfall)**
2. **Justificación de la Elección:**

- **Requisitos Fijos y Conocidos:** Los requisitos están documentados y se basan en un sistema existente, lo que permite definirlos de forma clara al inicio. El objetivo es replicar funcionalidad con mejoras mínimas.
- **Prioridad en la Exactitud:** La naturaleza secuencial del Cascada, con revisiones y aprobaciones en cada fase, asegura que la replicación de funcionalidades y la migración de datos se realicen con la máxima precisión.
- **Presupuesto y Cronograma Fijos:** La estructura predecible del Cascada, si los requisitos son estables, facilita la planificación y el control de costos y plazos, minimizando desviaciones.

3. Riesgos Potenciales y Mitigación:

- **Riesgo 1: Incompatibilidad de Datos:** Problemas en la migración de datos del sistema legado al nuevo formato.
 - **Mitigación:** Realizar un análisis exhaustivo de los datos existentes, mapeo de datos detallado y pruebas de migración de datos tempranas y exhaustivas en un entorno de pruebas.
- **Riesgo 2: Falta de Comprensión del Legado:** Aunque esté documentado, puede haber "conocimiento tácito" del sistema legado que no esté explícito.
 - **Mitigación:** Incluir a usuarios clave y expertos del sistema legado en las fases de requisitos y diseño, y realizar sesiones de extracción de conocimiento.

4. Modelo Alternativo Descartado: Modelo Ágil

- **Argumento 1:** El Ágil está diseñado para adaptarse a cambios y requisitos emergentes, lo cual es innecesario y potencialmente contraproducente para un proyecto con requisitos ya definidos y un objetivo de replicación.
- **Argumento 2:** La flexibilidad del Ágil podría llevar a desviaciones del objetivo principal (migración precisa y fiel) al introducir mejoras o nuevas funcionalidades no planificadas, lo que podría afectar el cronograma y la continuidad del negocio.