

# **Caso de Estudio: Necesidades de Capacitación para la Integración de Nuevos Desarrolladores en el Departamento de TI de la Universidad Adventista de Bolivia**

## **A. Portada**

**Empresa:** Universidad Adventista de Bolivia (UAB)

**Autor:** Juan Daniel Mamani Antezana

**Fecha:** 07 – 05- 2025

## **B. Resumen Ejecutivo**

Este caso de estudio presenta una problemática real observada dentro del entorno institucional de la Universidad Adventista de Bolivia (UAB): la integración deficiente de nuevos desarrolladores al equipo del Departamento de Tecnología de la Información (TI), un área vital para el sostenimiento y avance del ecosistema tecnológico universitario.

A través de una metodología cualitativa, se identificaron limitaciones clave en el proceso de onboarding de practicantes y pasantes: falta de documentación técnica, conocimientos técnicos insuficientes, ausencia de procesos formales de integración, y carencia de seguimiento. Para abordar esta situación, se diseñó y ejecutó un programa piloto basado en cuatro ejes: capacitación técnica, mentoría individual, documentación estructurada y evaluaciones continuas.

Los resultados del programa, implementado durante un semestre académico, fueron muy positivos: reducción del tiempo de adaptación, disminución de errores técnicos, mejora en la calidad del código, y mayor satisfacción por parte de los practicantes. Este caso propone un modelo de formación replicable, escalable y alineado con las necesidades de instituciones educativas que desean fortalecer sus equipos de desarrollo.

## **C. Introducción**

El desarrollo de software en instituciones educativas ha pasado de ser una actividad secundaria a convertirse en una necesidad estratégica. La Universidad Adventista de Bolivia (UAB), consciente de este cambio, ha decidido apostar por el fortalecimiento de su infraestructura digital a través de un Departamento de TI interno, el cual gestiona sistemas críticos como el académico, financiero y logístico.

### **1. Contexto institucional**

El Departamento de Tecnología de la Información fue creado hace aproximadamente dos años con el objetivo de atender las necesidades informáticas de las diferentes áreas académicas y administrativas. A pesar de su reciente fundación, ya lidera el desarrollo de un sistema académico propio basado en tecnologías como C# (lenguaje de backend), SQL Server (base de datos relacional) y Microsoft Report Builder (para generación de reportes institucionales).

Dado el creciente número de requerimientos y limitaciones presupuestarias, el departamento ha recurrido a estudiantes en práctica profesional, quienes representan una fuerza laboral clave pero desafiante, dado que su nivel técnico varía ampliamente.

## 2. Objetivo del estudio

El presente caso busca analizar de forma profunda las limitaciones observadas en la incorporación de estos nuevos talentos, evaluar su impacto en la calidad del producto desarrollado, y proponer soluciones efectivas que puedan ser institucionalizadas. El enfoque del estudio no solo es técnico, sino también organizacional y humano.

## D. Descripción del Problema

### 1. Situación inicial

Durante el año previo a la implementación del plan de capacitación, se detectaron los siguientes síntomas repetitivos:

- Los nuevos desarrolladores presentaban una **curva de aprendizaje lenta**, debido al desconocimiento de las herramientas utilizadas en los proyectos de la UAB.
- La **ausencia de manuales técnicos**, flujos documentados y reglas de codificación internas dificultaba la autonomía de los nuevos miembros.
- La **dependencia de conocimiento oral** implicaba inconsistencias en la información recibida por cada nuevo ingreso.
- El **uso incorrecto de herramientas como Git, Report Builder o Visual Studio** causaba errores críticos, como sobrescrituras de reportes o conflictos de versiones.
- La **falta de claridad en las tareas asignadas** producía ansiedad, estrés y desmotivación en los practicantes.

### 2. Consecuencias directas e indirectas

- **Retrasos en los cronogramas del sistema académico.**
- **Carga excesiva de los desarrolladores senior.**

- **Aumento del retrabajo y corrección de errores.**
- **Frustración entre los líderes técnicos.**
- **Rotación anticipada de practicantes.**
- **Quejas de usuarios por errores funcionales.**

### **3. Impacto organizacional**

El problema de integración afectaba transversalmente a la institución. No solo ponía en riesgo el avance del sistema académico, sino que también generaba desconfianza en áreas como registro académico, secretarías de facultades y administración general. Se hacía urgente establecer un modelo de incorporación que garantizara el éxito de los desarrolladores sin afectar la operación institucional.

### **E. Metodología**

La metodología utilizada en este caso fue de tipo **cualitativo, descriptivo y aplicado**, orientado a resolver un problema real con una solución práctica, documentada y transferible.

#### **Herramientas y técnicas aplicadas:**

1. **Observación directa:** Participación activa del autor del estudio en el entorno de desarrollo durante seis meses.
2. **Entrevistas semiestructuradas:** A 3 desarrolladores senior, 5 practicantes y 1 jefe de proyectos.
3. **Encuesta anónima:** Aplicada a 15 integrantes del equipo, evaluando percepción, frustraciones y sugerencias.
4. **Análisis FODA:** Para identificar debilidades, amenazas, fortalezas y oportunidades del proceso actual.
5. **Revisión documental:** Commits de Git, reportes de error, tareas Trello y códigos fuente del sistema.

### **F. Análisis de Datos**

#### **Resultados destacados de la investigación:**

- **90% de los nuevos desarrolladores** indicaron que no recibieron inducción formal.

- **80% aprendieron por prueba y error**, dependiendo de preguntas informales a compañeros.
- **75% considera que la documentación habría reducido sus errores.**
- **60% cometió errores graves en producción durante sus primeras semanas.**
- **50% se sintió desmotivado en su primer mes debido a la confusión y falta de guía.**

#### **Causas identificadas (por frecuencia - análisis de Pareto):**

1. Falta de capacitación técnica inicial (35%)
2. Ausencia de documentación formal (25%)
3. Deficiencia en comunicación y coordinación (15%)
4. Mala planificación de tareas (10%)
5. Poca autoconfianza del practicante (10%)
6. Falta de seguimiento o evaluación estructurada (5%)

Estas seis causas cubren el 100% de los problemas detectados, lo cual demuestra que el problema es sistémico, no aislado.

### **G. Solución Implementada**

#### **1. Capacitación intensiva**

Un curso de tres semanas para todo nuevo ingreso, dividido en:

- **Semana 1:** Lógica de backend, estructuras de código, introducción a C#.
- **Semana 2:** SQL Server, modelado de datos, consultas y procedimientos almacenados.
- **Semana 3:** Diseño de reportes con Report Builder y estándar de impresión institucional.

#### **2. Manual del Desarrollador**

Un documento compartido digitalmente que incluye:

- Diagramas UML del sistema académico.
- Flujos de trabajo y casos de uso por módulo.
- Ejemplos de commits bien documentados.
- Buenas prácticas de codificación, formateo, nomenclatura, errores comunes.

- Protocolo de despliegue seguro.

### 3. Mentoría técnica

Cada nuevo ingreso fue emparejado con un mentor que:

- Revisaba su código.
- Lo orientaba sobre procesos internos.
- Realizaba sesiones semanales de revisión.
- Levantaba alertas de aprendizaje lento o bloqueos.

### 4. Evaluaciones periódicas

- **Pruebas semanales** con feedback técnico.
- **Lista de cotejo** sobre participación, lógica, orden y actitud.
- **Evaluación 360°** entre mentor, jefe y compañero.

## H. Resultados

### Indicadores Cuantitativos

Indicador	Antes	Después	Mejora
Tiempo de adaptación promedio	3 semanas	1.5 semanas	-50%
Calidad de código entregado	60%	85%	+25%
Errores técnicos por sprint	15	9	-40%
Satisfacción del practicante	60%	90%	+30%
Carga de tiempo de los líderes (%)	30%	15%	-50%

### Comentarios cualitativos:

“Me sentí mucho más acompañado en este proceso que en otras prácticas que tuve antes.”

“El manual me ayudó muchísimo. No tuve que preguntar cosas básicas y me dio seguridad.”

“Los errores se redujeron porque ahora se revisa todo antes de subir.”

## I. Evaluación Crítica

**Fortalezas del proyecto:**

- Resultados tangibles en corto plazo.
- Mejora del clima laboral.
- Modelo replicable a otros equipos.
- Reducción de carga y estrés en personal senior.

**Debilidades:**

- Dependencia de mentores voluntarios.
- Requiere actualización continua de documentación.
- No existe plataforma de formación digital (LMS).

**Riesgos:**

- Descontinuidad si el liderazgo cambia.
- Saturación de mentores si ingresan muchos practicantes simultáneamente.
- Desmotivación si no se institucionaliza el reconocimiento.

**J. Conclusiones**

La incorporación efectiva de nuevos desarrolladores no es un proceso espontáneo. Exige planificación, acompañamiento, estándares técnicos y documentación clara. El modelo diseñado en la UAB demuestra que, con herramientas adecuadas y compromiso institucional, se puede lograr una integración eficiente, humana y productiva. Este enfoque no solo beneficia a los practicantes, sino que fortalece al equipo en su conjunto, contribuyendo a una cultura de mejora continua.

**K. Recomendaciones**

1. Formalizar el programa como política interna obligatoria.
2. Designar un responsable de capacitación técnica semestralmente.
3. Crear un repositorio centralizado de manuales, ejemplos y scripts.
4. Migrar el contenido del curso a una plataforma Moodle o LMS interna.
5. Promover sesiones de retroalimentación abierta para el equipo de TI.
6. Incorporar a expracticantes exitosos como mentores voluntarios.

## **L. Anexos (sugeridos)**

- Transcripciones de entrevistas.
- Capturas de pantallas de commits erróneos.
- Ejercicios usados en la capacitación.
- Resultados de encuestas de satisfacción.
- Fragmentos del Manual del Desarrollador.
- Flujos BPMN del proceso de onboarding.
- Cronograma de implementación del plan piloto.

## **M. Referencias Bibliográficas**

- Microsoft Corporation. "Microsoft SQL Server Documentation."  
<https://docs.microsoft.com/sql/>
- Troelsen, A. (2021). *Pro C# 9 with .NET 5*. Apress.
- Microsoft. "Report Builder documentation."  
<https://docs.microsoft.com/sql/reporting-services/report-builder/>
- Sommerville, I. (2016). *Software Engineering* (10th ed.). Pearson.
- Yin, R. K. (2017). *Case Study Research and Applications: Design and Methods*. SAGE Publications.