

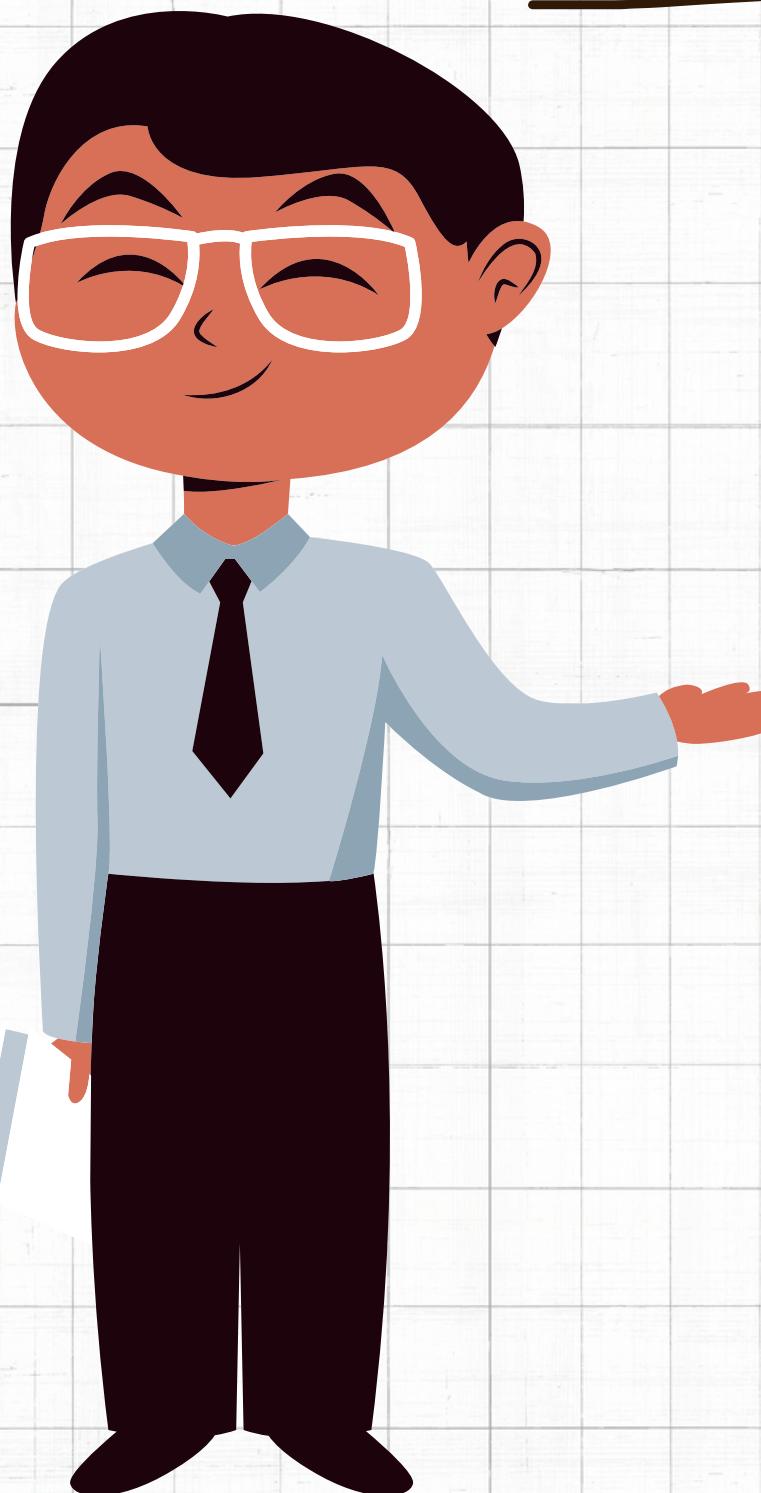
MODELO DE

DESARROLLO DE SOFTWARE

JUAN DAVID GARCIA
HECTOR
MAYRA TAMAYO



¿QUE SON LOS MODELOS DE DESARROLLO ?



Un modelo de desarrollo de software es una guía sistemática para crear software de manera eficiente. Define roles, etapas y entregables del proceso de desarrollo.

CARACTERISTICAS

Estructura y secuencia de actividades

Establecen una serie de actividades en un orden específico, como la recopilación de requisitos, diseño, implementación, pruebas y despliegue.

Iterativo e incremental

Muchos modelos de desarrollo de software utilizan un enfoque iterativo e incremental, dividiendo el proceso en ciclos repetitivos para agregar funcionalidad y mejorar aspectos existentes del software.

Colaboración y comunicación:

Los modelos de desarrollo de software fomentan la colaboración y la comunicación efectiva entre los miembros del equipo de desarrollo, los clientes y otras partes interesadas.

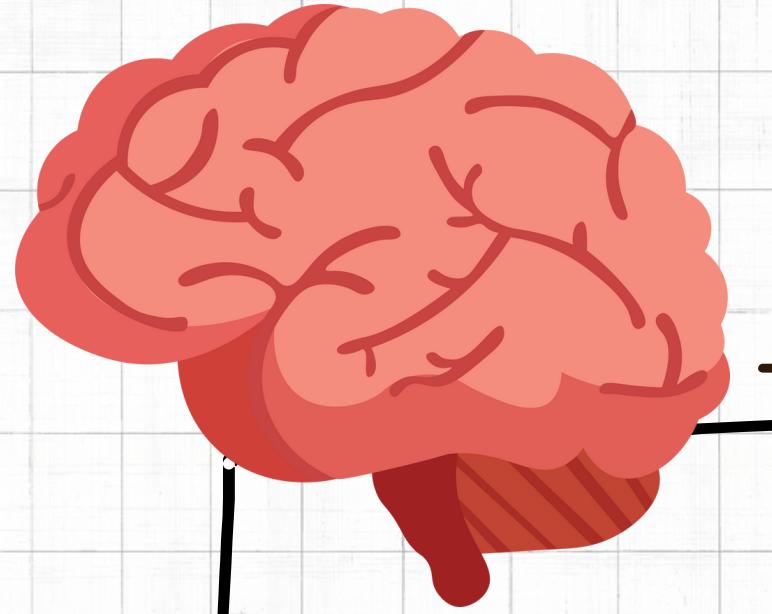
Gestión del riesgo

Los modelos de desarrollo de software abordan la gestión del riesgo mediante la identificación de posibles problemas y la implementación de estrategias para mitigarlos, incluyendo la planificación de contingencias, pruebas exhaustivas y medidas de seguridad.

MODELO EN ESPIRAL

DESARROLLO DE SOFTWARE

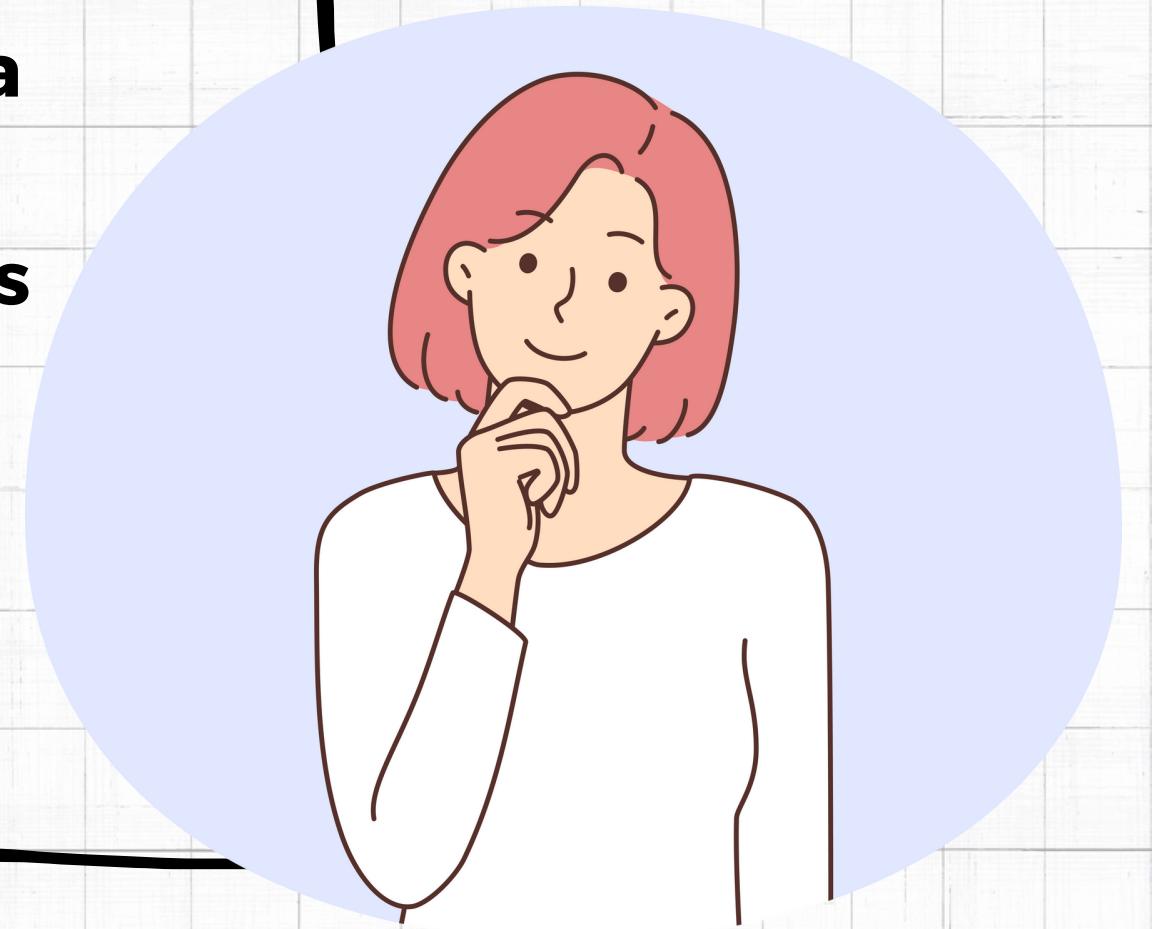




¿QUE ES ?

Modelo en Espiral del Ciclo de Vida del Software

El modelo en espiral del ciclo de vida del software es un enfoque iterativo e incremental para el desarrollo de software. A diferencia de otros modelos, como el modelo en cascada, el modelo en espiral permite la retroalimentación continua y la adaptación a medida que se desarrolla el software. A continuación, se presentan las características principales de este modelo:



Iterativo e incremental

se basa en ciclos repetitivos de planificación, análisis de riesgos, desarrollo y evaluación.

Enfoque basado en riesgos

se centra en la identificación y mitigación de riesgos.

Enfoque orientado a la calidad

El modelo en espiral se centra en la calidad del software.

CARACTERISTICAS

Flexibilidad y adaptabilidad

modelo en espiral permite cambios y ajustes a lo largo del proceso de desarrollo.



CASO PRÁCTICO DEL MÓDULO EN ESPIRAL

Imaginemos que una empresa de desarrollo de software ha sido contratada para crear un sistema de gestión de inventario para una tienda minorista. Utilizarán el modelo en espiral para desarrollar el software.

1

Planificación: En esta etapa, se definen los objetivos del proyecto, se identifican los requisitos del sistema y se establece un plan de desarrollo inicial.

2

Análisis de riesgos: Se realiza un análisis exhaustivo de los posibles riesgos asociados con el proyecto.

3

Desarrollo e implementación: Se desarrolla una versión inicial del software que cumple con los requisitos identificados.

4

Evaluación: Se evalúa la versión inicial del software para identificar posibles problemas y áreas de mejora.

5

Planificación del siguiente ciclo: Con base en los resultados de la evaluación, se realiza una planificación para el siguiente ciclo.

6

Repetición de los ciclos: Los ciclos de planificación, análisis de riesgos, desarrollo e implementación, y evaluación se repiten hasta que se logre la calidad deseada y se cumplan todos los requisitos del sistema.

VENTAJAS

Reduce riesgos del proyecto

Incorpora objetivos de calidad

Integra el desarrollo con el mantenimiento



DESVENTAJAS

Es un modelo que genera mucho trabajo adicional

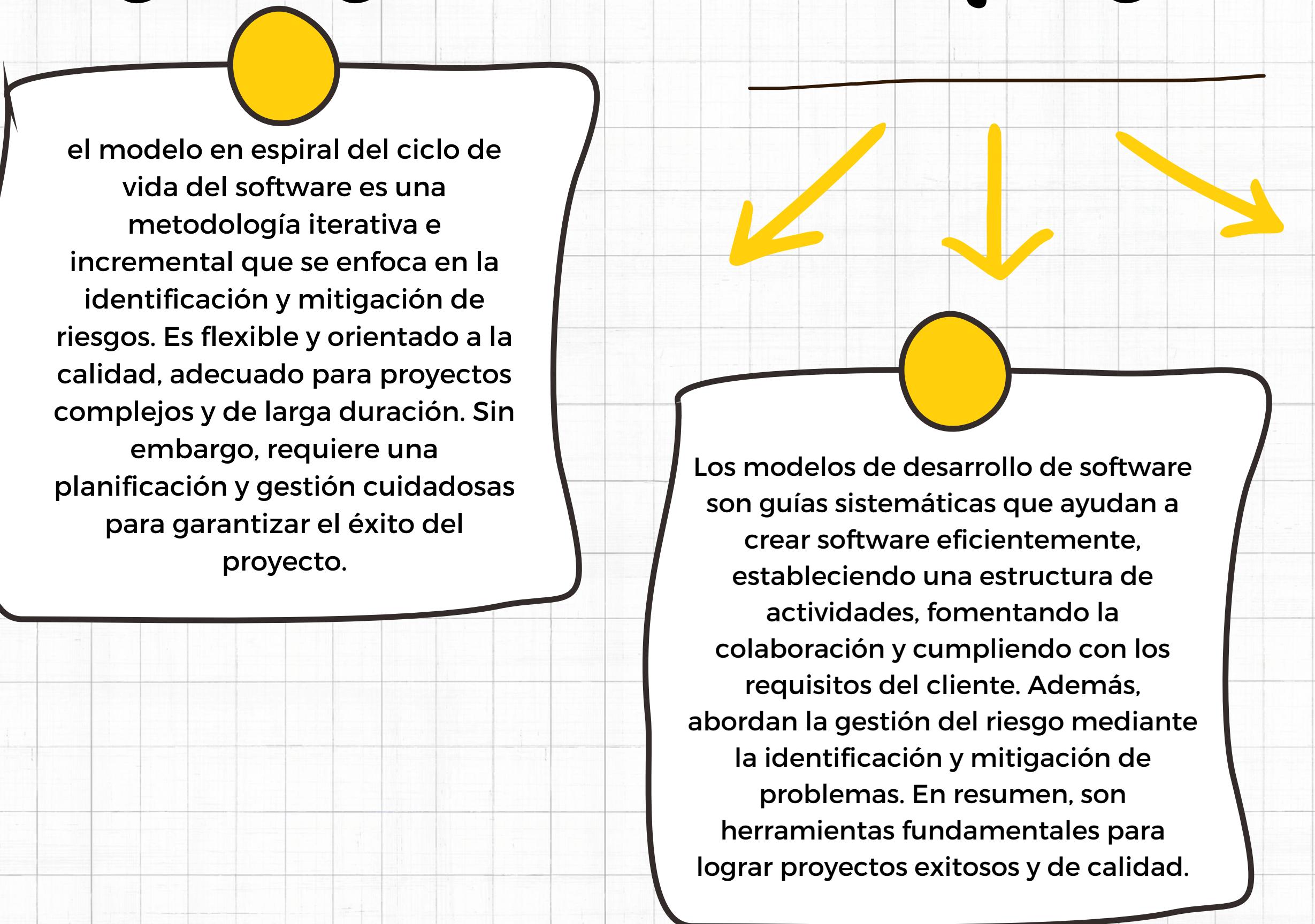
Costos más altos

Requiere experiencia



CONCLUSIONES Y RECOMENDACIONES

el modelo en espiral del ciclo de vida del software es una metodología iterativa e incremental que se enfoca en la identificación y mitigación de riesgos. Es flexible y orientado a la calidad, adecuado para proyectos complejos y de larga duración. Sin embargo, requiere una planificación y gestión cuidadosas para garantizar el éxito del proyecto.



Los modelos de desarrollo de software son guías sistemáticas que ayudan a crear software eficientemente, estableciendo una estructura de actividades, fomentando la colaboración y cumpliendo con los requisitos del cliente. Además, abordan la gestión del riesgo mediante la identificación y mitigación de problemas. En resumen, son herramientas fundamentales para lograr proyectos exitosos y de calidad.

- Realizar evaluaciones de riesgos periódicas y tomar medidas para mitigarlos.
- Establecer una comunicación clara y continua con los usuarios y las partes interesadas para recopilar retroalimentación y realizar mejoras.
- Contar con un equipo experimentado en la identificación y gestión de riesgos.
- Establecer un plan de proyecto claro y flexible que se pueda adaptar a los cambios y requisitos cambiantes.

BIBLIOGRAFÍAS

Pressman, R. S.
(2014). Ingeniería del
software: un
enfoque práctico.
McGraw-Hill
Education.

Sommerville, I.
(2016). Ingeniería de
software. Pearson
Educación.